STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA

July 1965

Report 4

FLUX SWITCHING IN MULTIPATH CORES

Prepared for:

JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CALIFORNIA

CONTRACT 950943 UNDER NAS7-100
(Extension of work under
Contract 950095 under NASw-6)

By: D. NITZAN V. W. HESTERMAN

SRI Project 5094

Approved: D. R. BROWN, MANAGER
COMPUTER TECHNIQUES LABORATORY

J. D. NOE, EXECUTIVE DIRECTOR
ENGINEERING SCIENCES AND INDUSTRIAL DEVELOPMENT

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100.

Copy No. 67

ABSTRACT

In the total $\dot{\phi}(t)$ waveform of a ferrite core switched by mmf, $F(t)$, from negative remanence, $\phi = -\phi_r$, three components are distinguished: elastic, decaying inelastic, and main inelastic. The model for the elastic component, $\dot{\phi}_\epsilon$, is $\dot{\phi}_\epsilon = \epsilon\dot{F}$, where ϵ is a function of F, ϕ, and the previous switching. The model for the decaying inelastic component, $\dot{\phi}_i$, is $\dot{\phi}_i = \lambda_i(F - F_i)^{\nu_i} \exp[-(t - T_i)(F - F_i)/C_i]$, where λ_i, F_i, ν_i, and C_i are switching parameters. The model for the main inelastic component, $\dot{\phi}_{ma}$, is $\dot{\phi}_{ma} = \dot{\phi}_p\{1 - [(2\phi + \phi_r - \phi_d)/(\phi_r + \phi_d)]^2\}$, where $\dot{\phi}_p$ is the peak value of $\dot{\phi}$ and ϕ_d is the ϕ value on the static $\phi(F)$ curve, both of which are given as functions of F. If F is large (compared with the coercive mmf, F_c), then $\dot{\phi}_i + \dot{\phi}_{ma} \cong \dot{\phi}_p\{1 - [(2\phi + \phi_s - \phi_d)/(\phi_s + \phi_d)]^2\}$, where ϕ_s is saturation flux. A computer program is written and applied in computing $\dot{\phi}(t)$ and its components for a thin toroidal ferrite core. The results agree well with experimental $\dot{\phi}(t)$ waveforms obtained by using $F(t)$ with different rise times ($T_r \cong 0.1$ μsec and $T_r \cong 0.02$ μsec) and different amplitudes (seven to eight values, varying from 2/3 tc more than twice F_c). A small delay (of the order of $0.15\ T_r$) between computed and experimental $\dot{\phi}_\epsilon(t)$ shows that an improved model for the elastic $\dot{\phi}$ is $\delta\ddot{\phi}_\epsilon + \dot{\phi}_\epsilon = \epsilon\dot{F}$, where δ is a constant proportional to the viscous damping. The parameters λ_i and C_i were found to be affected by T_r: as T_r decreases, λ_i increases (slightly) and C_i decreases. The flux change involved in obtaining the static $\phi(F)$ curve for low F values is much larger than $\int_0^\infty \dot{\phi}_i\,dt$; this flux change is ascribed to a very slow switching component which is either part of $\dot{\phi}_i$ or $\dot{\phi}_{ma}$ with threshold lower than F_0''.

Numerical analyses and computer programs are given for three additional magnetic circuits: a loaded core, a core-diode-transistor binary counter, and a loaded, saturable, three-leg core. In each case, the basic problem is to solve a set of first-order nonlinear differential equations together with a transcendental solution for some of the time variables. If the load is inductive, the loaded-core program provides more exact results that agree better with experimental data than those obtained previously. The

binary-counter program includes computation *vs.* time of three currents and F, $\dot{\phi}$, and ϕ of each of two coupled cores during the fast-switching mode of operation of a single stage in the counter. The computed results are essentially identical with results computed by a more complex and more exact method of solution of differential equations (Runge-Kutta and Adams) and agree quite well with experimental data. The last program includes three types of computation for flux division: time variables (F, $\dot{\phi}$, and ϕ of each leg), flux-division ratio D *vs.* drive amplitude for different loads, and D *vs.* leg-length ratio for different loads and drive amplitude. The agreement between computed and measured D *vs.* NI is satisfactory except for very low values of drive mmf.

The switching properties of a thin-ring core in response to a ramp drive, $F = kt$, have been investigated for k varying from 0.1 to 10 amp-turn/μsec. Experimental $\dot{\phi}_p(k)$ and $t_p(k)$ curves are compared with computed curves based on the parabolic model $\dot{\phi} = \lambda(F - F_0'')^{\nu}[1 - (2\varphi + \varphi_s - \varphi_d)^2/(\varphi_s + \varphi_d)^2]$. The experimental $\dot{\phi}_p(k)$ curve fell entirely below the computed curve when step-F parameters were used in the computation. The computed $\dot{\phi}_p(k)$ curve could be made to agree with the experimental curve over the entire range of k by using a lower value of λ, λ_r. The computed and experimental $t_p(k)$ curves could be made to agree at any k value by using a lower value of F_0'', F_{0r}''. However, exact agreement could be obtained at only one k value because these $t_p(k)$ curves cross each other. In spite of this problem, the above model can be used for practical applications if k does not vary over a wide range.

A study was made of the effect of temperature (in the range $-50°C$ to $+75°C$) on the switching properties of two square-loop ferrite toroids (a thin ring and a thick ring). Temperature coefficients have been found for the parameters of the model described above. The static $\phi(F)$ curves for a partially set state changed with temperature by approximately the same percentage as the major static $\phi(F)$ curves. As the temperature increased, the $\dot{\phi}_p(F)$ curves for a step-F drive shifted to lower F values but their slope and curvature were hardly affected. The $\dot{\phi}_p(k)$ curves for a ramp-F drive for different temperatures cross each other. The $t_p(k)$ curves were shifted downward by an increase in temperature, but were unaffected in slope and curvature. These effects are mostly due to temperature variations in F_{0r}'' and λ_r.

CONTENTS

ILLUSTRATIONS

TABLES

LIST OF SYMBOLS

Symbol	Definition	Reference
A	Cross-sectional area of a leg	Rep. 1, Fig. 19
B	Flux density	
B_r	Maximum residual B	Rep. 2, Fig. 2
B_s	Saturation B	
$\dot{B}$	Time rate of change of B	
$\dot{B}_i$	Decaying inelastic $\dot{B}$	Eq. (13), p. 10
C	Capacitance	
C	$\tanh^{-1}(\phi_r/\phi_s)$ in derivation of $\dot{\phi}_p(k)$ and $t_p(k)$ for ramp-F switching	Eq. (101), p. 93; Eq. (102), p. 94
C_i	Parameter in a model for decaying inelastic $\dot{\phi}$	Eq. (5), p. 7
E_k	Parameter in a model for a junction diode	Eq. (42), p. 46
F	MMF	
F_B	MMF at boundary between nonlinear and linear regions of $\dot{\phi}_p(F)$	Rep. 3, Fig. 3
F_c	Coercive F	Rep. 3, Fig. 1
F_D	Amplitude of drive MMF	Rep. 3, Fig. 6
F_{DB}	Boundary value for F_D below which $T_{ip} = T_r$	Eq. (12), p. 9
F_d^n	Static F threshold	Rep. 3, Fig. 1
F_{ex}	Excess F above threshold	
F_i	MMF threshold for decaying inelastic $\dot{\phi}_i$	Eq. (17), p. 10
F_0	MMF threshold obtained by extrapolating linear $\dot{\phi}_p(F)$ to F axis	Rep. 3, Fig. 3; Eq. (19), p. 11

Symbol	Definition	Reference
F_0''	Dynamic F threshold	Rep. 3, Fig. 3; Eq. (19), p. 11
F_{0r}''	F_0'' of ramp-F switching	p. 96; Eq. (112), p. 102
F_{0i}	Asymptotic value for F_i	Fig. 5, p. 11
F_{12}	Boundary between first and second regions of computed static $\phi(F)$ curve, *i.e.*, $H_q l_i$	Rep. 2, pp. 6 and 28
F_{23}	Boundary between second and third regions of computed static $\phi(F)$ curve, *i.e.*, $H_q l_o$	Rep. 2, pp. 6 and 28
$\dot{F}$	Time rate of change of F	
H	Magnetic field	
H_a	Parameter in the hyperbolic model for static $B(H)$	Rep. 2, Eq. (10) and Fig. 2
H_B	H at boundary between linear and non-linear regions of $\dot{B}_p(H)$	Rep. 2, Fig. 15
H_c	Coercive H	Rep. 2, Eq. (14) and Fig. 2
H_i'	H threshold for decaying inelastic $\dot{B}$	Eq. (15), p. 10
H_n	Parameter in the hyperbolic model for static $B(H)$	Rep. 2, Eq. (11) and Fig. 2
H_q	Parameter in the hyperbolic model for static $B(H)$	Rep. 2, Eq. (11) and Fig. 2
H_0	H threshold, obtained by extrapolating linear $\dot{B}_p(H)$ to H axis	Rep. 2, Eq. (24) and Fig. 15
H_0''	Dynamic H threshold	Rep. 2, Eq. (83) and Fig. 15
$\overline{H}$	Average H	
h	Height of a leg (or core)	Rep. 1, Fig. 19
I	Drive-current amplitude in flux-division experiment	Fig. 24, p. 73
I_c	Asymptotic value of collector current in binary counter	Fig. 17, p. 52

Symbol	Definition	Reference
ID	Inside diameter of a toroid	
I_D	Amplitude of drive current	Rep. 3, Fig. 6
I_s	Initial value of i_s in a binary counter	Fig. 17, p. 52; Eq. (72), p. 62
I_0	Saturation current in a model for a junction diode	Eq. (42), p. 46
i	Instantaneous drive current in flux-division experiment	Eq. (93), p. 77
i_c	Collector current in a binary counter	Fig. 17, p. 52
i_D	Drive current	Rep. 3, Fig. 14
i_d	Loop current via diode in a binary counter	Fig. 19, p. 57
i_L	Current source in a binary counter	Fig. 19, p. 57
i_L	Load current	Rep. 3, Fig. 14
i_s	Drive current in a binary counter during STORE count	Fig. 17, p. 52; Fig. 19, p. 57
i_{sr}	Drive current in a binary counter during CARRY count	Fig. 17, p. 52
J	Index of iteration in a numerical solution of variables for each nth Δt, e.g., $\phi_{n(j)}$, $\dot{\phi}_{n(j)}$, $q_{n(j)}$, $\dot{q}_{n(j)}$, $\dot{\phi}_{n(j-1)}$, etc.	
k	Slope of ramp $F(t)$	Eq. (106), p. 99; Fig. 32, p. 100
L	Inductance	
l	Average leg length	Rep. 3, Fig. 5
l_i	Length of the short edge of a leg	Rep. 3, Fig. 5
l_o	Length of the long edge of a leg	Rep. 3, Fig. 5
M_i	Material parameter in a model for decaying inelastic $\dot{B}$	Eq. (16), p. 10
N	Number of turns	
N_c	N of load winding	Rep. 3, Fig. 14

Symbol	Definition	Reference
N_D	N of drive winding	Rep. 3, Fig. 14
N_L	N of load winding in flux-division experiment	Fig. 24, p. 73
NV	Negligible Value of $\dot{\phi}_\epsilon$	p. 16
n	Index number for a time element Δt and the associated variables, *e.g.*, ϕ_n, $\dot{\phi}_n$, etc.	*e.g.*, p. 22
OD	Outside diameter of a toroid	
P	Ratio of computed $\dot{\phi}_p$ [using static $\phi(F)$-limited parabolic model] and experimental $\dot{\phi}_p$ for ramp-F drive	Eq. (108), p. 102
q	Electric charge	Eq. (43), p. 46
$\dot{q}$	Load current, $i_L = dq/dt$	Eq. (43), p. 46
R	Resistance	
R_d	Forward resistance of a diode	Eq. (42), p. 46
R_L	Load resistance	Rep. 3, Fig. 14
r_i	Inside radius of a toroid	
r_o	Outside radius of a toroid	
S_p	Maximum slope of drive current	Fig. 9, p. 23
T	Duration of flux switching in flux-division experiment	Fig. 24, p. 73
T_i	Time when $\dot{\phi}_i$ begins to rise	Fig. 4, p. 9
T_{ip}	Time when $\dot{\phi}_i$ reaches peak	Fig. 4, p. 9
T_n	Abbreviation for $(13 - t \cdot 10^8)$ in i_L function in binary counter	Eq. (74), p. 64
T_r	Rise time of a drive pulse	Fig. 9, p. 23
T_{rB}	Boundary value for T_r below which $T_{ip} = T_r$	Eq. (10), p. 8; Fig. 4, p. 9

Symbol	Definition	Reference
t	Time	
t_m	Time when drive current reaches maximum slope	Fig. 9, p. 23
t_p	Peak time of $\dot{\phi}$	Rep. 3, Fig. 1
u	Drive-current parameter	Fig. 9, p. 23; Eq. (34), p. 24
V	Supply voltage in a binary counter	Fig. 17, p. 52
V_d	Forward voltage across a junction diode	Eq. [56(b)], p. 58
V_{min}	Minimum allowed V in binary counter	p. 55
V_1	Abbreviation for $(\phi_s - \phi_r)/[(l_o - l_i)H_a]$	Rep. 3, Eq. (28)
V_2	Abbreviation for $[(\phi_s + \phi_r)H_q]/[(l_o - l_i)H_n]$	Rep. 3, Eq. (29)
v_{R_1-L}	Voltage across R_1 and L in binary counter	Eq. (49), p. 56
w	Width of a leg	*e.g.*, Fig. 24, p. 73
y	General time variable	Eqs. (70) and (71), p. 61
$\dot{y}$	Time rate of change of y	Eqs. (70) and (71), p. 61
Δt	Small time increment used in computation of time variables	*e.g.*, Eq. (29), p. 22
Δt	The time interval used to measure the slope of a ramp-F drive	Fig. 32, p. 100
$\Delta\phi$	Change of ϕ	
$\Delta\phi_d$	$\Delta\phi$ associated with obtaining a point on static $\phi(F)$ curve, *i.e.*, $\phi_d + \phi_r$	Eq. (41), p. 41
$\Delta\phi_i$	$\Delta\phi$ due to $\dot{\phi}_i$	Eq. (38), p. 39
$\Delta\phi_{i(\infty)}$	$\Delta\phi_i$ for $t \to \infty$	Eq. (39), p. 39
$\Delta\phi_\epsilon$	Elastic $\Delta\phi$	Eq. (37), p. 39

Symbol	Definition	Reference
$\Delta\phi_{(\infty)}$	$\Delta\phi$ for $t \to \infty$	Eq. (40), p. 39
δ	Constant proportional to viscous damping in a model for $\dot{\phi}_\epsilon$	Eq. (35), p. 35
δi	Correction to i in iterative solution for i	p. 63
ϵ	Coefficient of elastic $\dot{\phi}$	Eqs. (2) and (3), p.4; Fig. 3, p. 5
ζ	Coefficient of inelastic $\dot{B}$	Rep. 2, Eq. (24)
ζ_p	Peak ζ	p. 76; Rep. 2, Eq. (82)
η	Flux form factor in parabolic model for main $\dot{\phi}$	Eq. (20), p. 11
κ	Coefficient of main inelastic $\dot{B}$ in nonlinear $\dot{B}_p(H)$ region	p. 76; Rep. 2, p. 38
κ_i	Coefficient of decaying inelastic $\dot{B}_i$	Eq. (14), p. 10
λ	Coefficient of main inelastic $\dot{\phi}$ in nonlinear $\dot{\phi}_p(F)$ region	Eq. (19), p. 11
λ_i	Coefficient of decaying inelastic $\dot{\phi}_i$	Eq. (5), p. 7
λ_r	λ for ramp-F switching	p. 96; Eq. (109), p. 102
ν	Power coefficient of main inelastic $\dot{\phi}$ in nonlinear $\dot{\phi}_p(F)$ region	Eq. (19), p. 11
ν_i	Power coefficient of decaying inelastic $\dot{\phi}_i$	Eq. (5), p. 7
ρ	Coefficient of inelastic $\dot{\phi}$	Rep. 1, Eq. (38)
ρ_i	Coefficient of decaying inelastic $\dot{\phi}_i$	Eq. (4), p. 6
ρ_p	Peak ρ	Eq. (19), p. 11
$\bar{\rho}$	Average ρ	Rep. 2, pp. 17-19
τ_s	Switching time	Eq. (90), p. 76
ϕ	Magnetic flux	

Symbol	Definition	Reference
ϕ_d	ϕ value on a static $\phi(F)$ curve	Rep. 3, Fig. 1
ϕ_r	Maximum residual ϕ	Rep. 3, Fig. 1
ϕ_s	Saturation ϕ	
ϕ'_d	$d\phi_d/dF$	Rep. 3, Eq. (31), (32), or (35)
$\dot{\phi}$	Time rate of change of ϕ	
$\dot{\phi}_i$	Decaying inelastic $\dot{\phi}$ component	p. 6
$\dot{\phi}_{ip}$	Peak $\dot{\phi}_i$	Eq. (11), p. 8; Fig. 4, p. 9
$\dot{\phi}_{ma}$	Main inelastic $\dot{\phi}$ component	p. 11
$\dot{\phi}_p$	Peak $\dot{\phi}$	Eq. (19), p. 11
$\dot{\phi}_\epsilon$	Elastic component of $\dot{\phi}$	p. 4
$\dot{\phi}'$	$\partial\dot{\phi}/\partial F$	p. 16
$\dot{\phi}'_i$	$\partial\dot{\phi}_i/\partial F$	Eq. (27), p. 17
$\dot{\phi}'_{ma}$	$\partial\dot{\phi}_{ma}/\partial F$	p. 17
$\dot{\phi}'_p$	$d\dot{\phi}_p/dF$	Rep. 3, Eq. (37) or (39)
$\dot{\phi}'_\epsilon$	$\partial\dot{\phi}_\epsilon/\partial F$	Eq. (25), p. 16
$\dot{\phi}^*$	$\partial\dot{\phi}_{ma}/\partial\phi$	Eq. (28), p. 18

ACKNOWLEDGMENTS

The assistance of D. F. Fraser in computer programming is appreciated.

We wish to thank Alan J. DeVilbiss for his comments relative to the analysis of the core-diode-transistor binary counter.

PREFACE

This project report, Contract 950943 under NAS7-100, Stanford Research Institute Project 5094, is an extension of work under a previous project, Contract 950095 under NASw-6, SRI Project 3696.

From an engineering viewpoint, the modeling of the terminal properties of ferrimagnetic or ferromagnetic core materials is useful. There are different ways to study these terminal properties. One extreme way is to study each property in as many core materials as possible before proceeding with the next property. Another extreme way is to study all the properties of one given core material before proceeding to the next core material. The present investigation lies between the extremes, although it is closer to the latter. In the past, we have studied the major characteristics of switching from a hard state ($\phi = -\phi_r$) and from a certain type of soft state ($|\phi| < \phi_r$) in a limited number of ferrite core materials. Among the materials studied, two magnesium-manganese ferrite materials have been investigated relatively thoroughly. This report describes the initial elastic and inelastic $\dot{\phi}$ spikes of one core material, the properties of ramp-F switching in three core materials, and the temperature effect on step-F switching, ramp-F switching and static $\phi(F)$ curves in two core materials. Additional properties of these materials need to be investigated, but other materials also need investigation.

In Report 3, the switching model was used in computational analyses* of magnetic circuits (unloaded core, loaded core, and core-diode shift register). The agreement with experimental data has encouraged us to extend this application to other magnetic circuits. Consequently, in this report, computational analyses and experimental verification are given for the initial $\dot{\phi}(t)$ spikes of an unloaded core, for a loaded core (using an improved algorithm), for a core-diode-transistor binary counter, and for flux division in a loaded saturable three-leg core.

*By "computational analysis" we mean a numerical analysis which is programmed and run on a digital computer.

I ELASTIC AND INELASTIC FLUX-SWITCHING COMPONENTS

A. INTRODUCTION

In this section, improved switching models are proposed and verified experimentally for three components of $\dot{\phi}$: the elastic $\dot{\phi}$ spike, $\dot{\phi}_\epsilon$, occurring while F changes in time; the decaying inelastic $\dot{\phi}$ components, $\dot{\phi}_i$, which falls exponentially after reaching peak during the rise of F; and the main inelastic $\dot{\phi}$ component, $\dot{\phi}_{ia}$, which is bell-shaped and, if F is not too low, accounts for most of the flux switching. The first and third components are well known, and were discussed in detail and applied in the previous three reports[1,2,3]* (hereinafter referred to as Reports 1, 2, and 3). The decaying $\dot{\phi}$ component, $\dot{\phi}_i$, was introduced in Report 3 (using the symbol $\dot{\phi}_{\rho i}$ instead of $\dot{\phi}_i$), where an attempt was made to model this component.

An additional experimental verification for the existence of $\dot{\phi}_i$ is given in Fig. 1 by showing $\dot{\phi}(t)$ waveforms of a thin toroidal ferrite core (Core E-6, Report 3, p. 23). These $\dot{\phi}(t)$ waveforms resulted from the

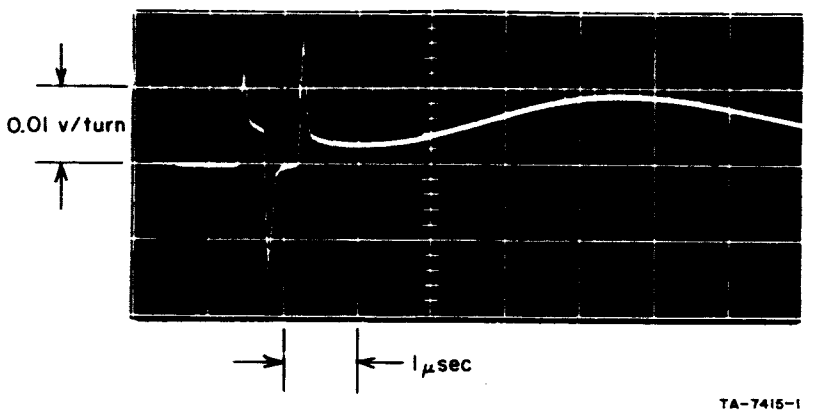


FIG. 1 $\dot{\phi}(t)$ OSCILLOGRAM OF INTERRUPTED-F SWITCHING IN A THIN TOROIDAL FERRITE CORE USING $F > F''_0$
Core E-6: OD/ID = 1.06; $F''_0 = 0.95$ amp-turn;
$F_c = 0.9$ amp-turn; $F = 1.17$ amp-turn.

* References are listed at the end of the report.

interruption of a step-F switching with low F ($F = 1.3\ F_c$, where F_c is the coercive mmf). The three fast-switching $\dot{\phi}$ spikes (the second of which is negative) have been retouched because the original traces were too faint for photographic reproduction. The elastic $\dot{\phi}$ spikes shown occur during the rise and fall of the first F pulse and during the rise of the second F pulse. The two F pulses have the same amplitude. During the first F pulse and in the beginning of the second F pulse, the total $\dot{\phi}(t)$ is decaying despite the rise of the main $\dot{\phi}(t)$ component. The difference between the total $\dot{\phi}(t)$ and the main $\dot{\phi}(t)$ component is the decaying component, $\dot{\phi}_i$. Our conclusion that $\dot{\phi}_i$ is due to inelastic domain-wall motion (*cf.* Report 3, p. 12) is verified by the following three observations:

(1) The relaxation time of $\dot{\phi}_i$ is much longer than typical relaxation time of rotation of magnetization.[4]

(2) Referring to the first F pulse, the area under $\dot{\phi}_i$ (following the rise time) is much larger than the area under the negative decaying $\dot{\phi}$ component (following the fall time). The latter is elastic in nature, and presumably results from the excess of the number of walls moving backward over the number of walls moving forward toward energy minima when F is suddenly interrupted. This excess in number of walls is due to the slower average velocity of the walls passing between energy-valley and energy-peak positions compared with walls moving between energy-peak and energy-valley positions.

(3) The $\dot{\phi}_i$ waveform that follows the positive $\dot{\phi}$ spike of the second F pulse continues to decay smoothly from its value at the end of the first F pulse. This behavior is characteristic only of domain-wall motion.

Experimental $F(t)$ and $\dot{\phi}(t)$ oscillograms of interrupted-F switching are shown in Fig. 2 for the same conditions as in Fig. 1, except that $F = 0.8$ amp-turn. Since $F < F_0''$ ($F_0'' = 0.95$ amp-turn), the main $\dot{\phi}(t)$ component is not present, and $\dot{\phi}(t) = \dot{\phi}_\epsilon(t) + \dot{\phi}_i(t)$.

Flux-switching models have been proposed for $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, and $\dot{\phi}_{ma}$ in Report 3 (the subscript ma is not added in Reports 1, 2, and 3). In the course of further investigation of $\dot{\phi}_i$, it was found that certain modifications should be incorporated into the models for both $\dot{\phi}_i$ and $\dot{\phi}_{ma}$ in order to obtain better agreement with experimental data. These modified models are given next. For completeness, we shall also summarize briefly the model for $\dot{\phi}_\epsilon$.

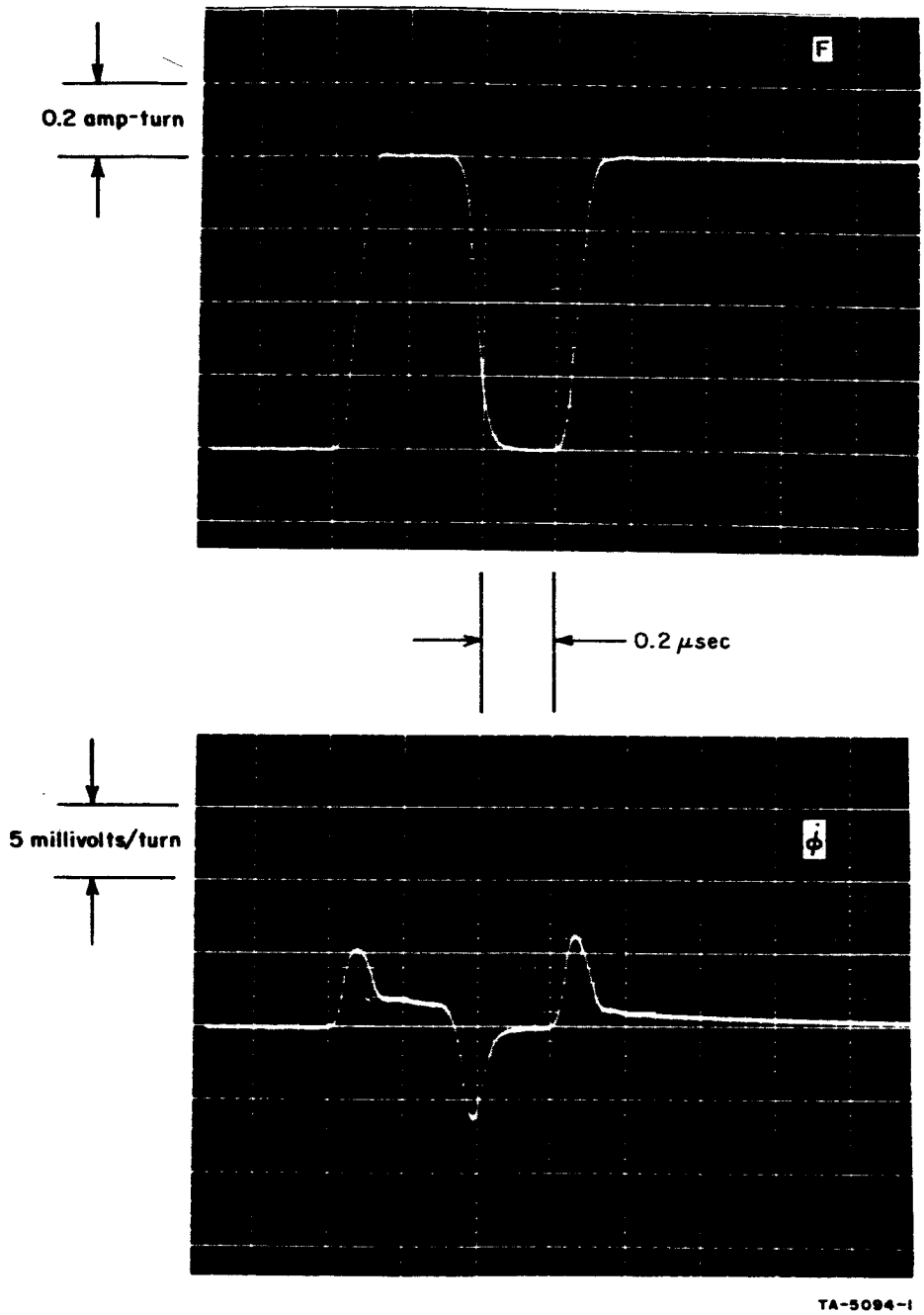


FIG. 2 F(t) AND $\dot{\phi}(t)$ OSCILLOGRAMS OF INTERRUPTED–F SWITCHING IN A THIN TOROIDAL FERRITE CORE USING $F < F_0''$
Core E-6: OD/ID = 1.06; $F_0'' = 0.95$ amp-turn; $F_c = 0.9$ amp-turn;
$F = 0.8$ amp-turn.

B. FLUX-SWITCHING MODELS

Consider a thin core (or leg) which is driven by $F(t)$. Based on the discussion above, the total $\dot{\phi}$ is expressed as

$$\dot{\phi} = \dot{\phi}_{\epsilon} + \dot{\phi}_{i} + \dot{\phi}_{ma} \quad . \tag{1}$$

Semiempirical models for $\dot{\phi}_{\epsilon}$, $\dot{\phi}_{i}$, and $\dot{\phi}_{ma}$ are reviewed and modified as follows.

1. Elastic $\dot{\phi}$ Spike

Following Eq. (32) in Report 1 (p. 23),

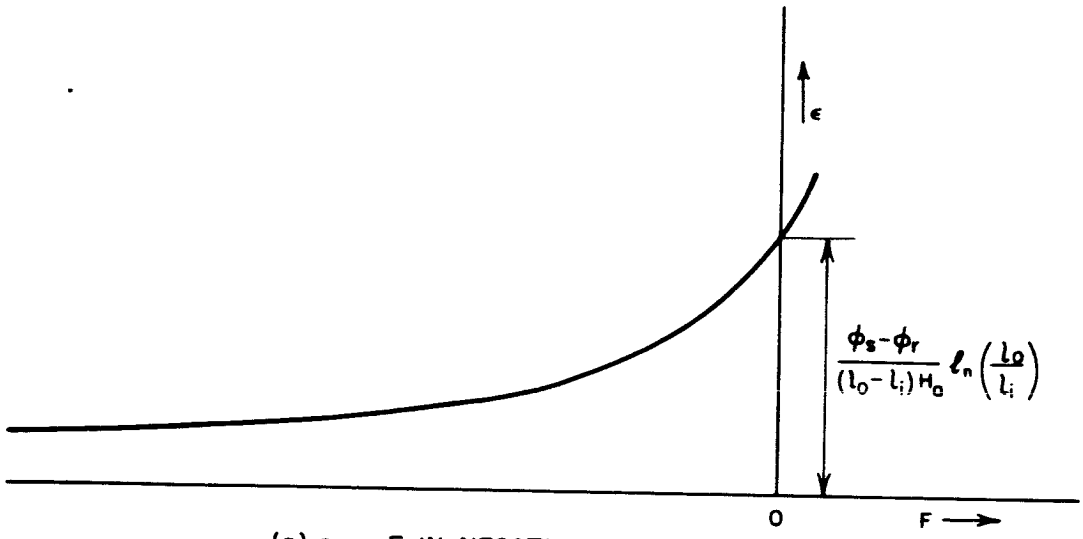
$$\dot{\phi}_{\epsilon} = \epsilon \dot{F} \quad , \tag{2}$$

where ϵ is a function of F, ϕ, and the history of previous switching. For a core in saturation, $\epsilon(F)$ is given by Eq. (34), Report 1, and is plotted in Figs. 26 and 27 of Report 1 [for convenience, it is replotted in Fig. 3(a)]. For $\phi = -\phi_r$ and if $|F|$ is not much larger than F_c, then for a toroidal core [*cf.* Eq. (7), Report 3, p. 10],

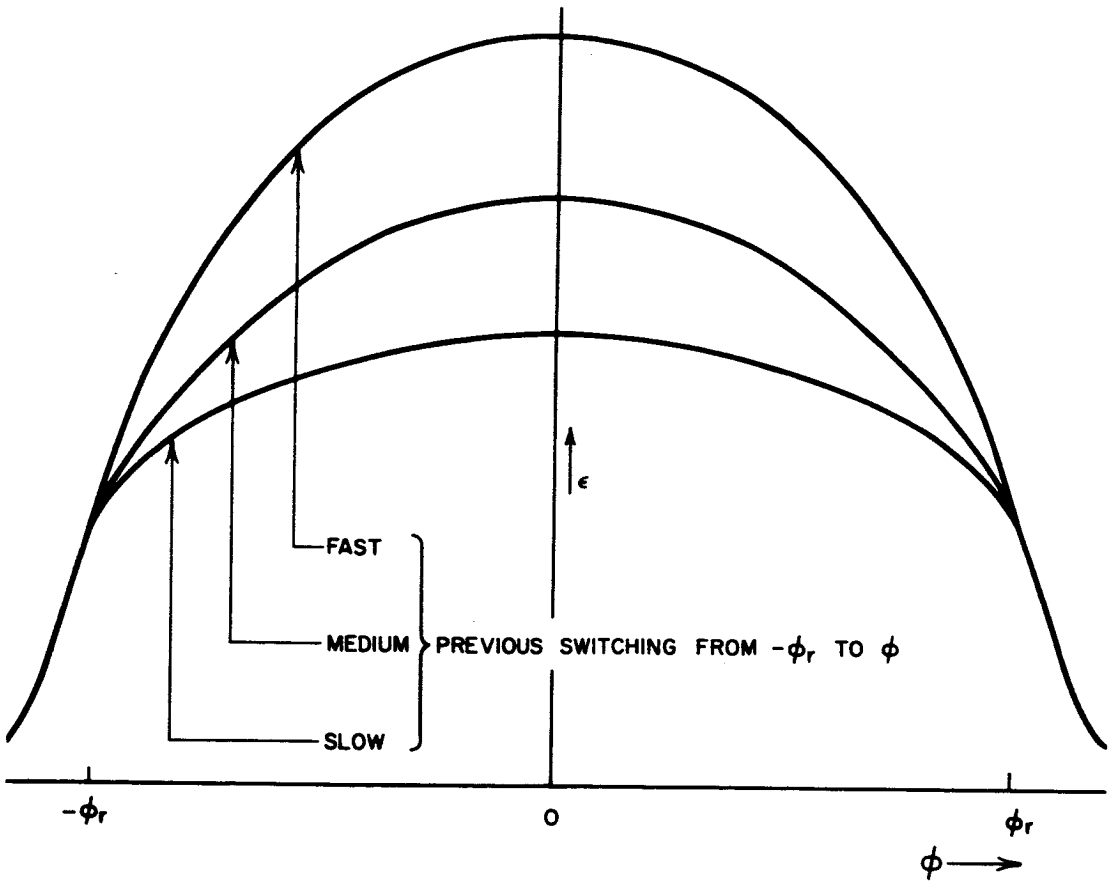
$$\epsilon \cong \frac{\phi_s - \phi_r}{2\pi(r_o - r_i)H_a} \ln\left(\frac{r_o}{r_i}\right) \quad , \tag{3}$$

where ϕ_s is saturation flux, ϕ_r is maximum residual flux, H_a is a material parameter, and r_o and r_i are outside and inside radii of the toroid.

The effects of ϕ and the previous switching (to reach ϕ) on ϵ are shown schematically in Fig. 3(b). For $|\phi| > \phi_r$, $\epsilon(\phi)$ may be derived from $\epsilon(F)$, shown in Fig. 3(a), and the static $\phi(F)$ curve (not shown). For $-\phi_r \leq \phi \leq \phi_r$, $\epsilon(\phi)$ peaks near $\phi = 0$; for a given ϕ value, the faster the previous switching from $-\phi_r$ to ϕ is, the higher is ϵ.[5] More investigation in the area of flux switching from a partially-set state is needed before we can propose a model for $\epsilon(\phi,$ switching history).



(a) ϵ vs. F IN NEGATIVE SATURATION



(b) ϵ vs. ϕ WITH PREVIOUS SWITCHING AS A PARAMETER

TA-5094-2

FIG. 3 EFFECTS OF F, ϕ, AND PREVIOUS SWITCHING ON ELASTIC-SWITCHING COEFFICIENT ϵ

2. Inelastic Decaying $\dot{\phi}$

a. INTRODUCTION

In Report 3, Eq. (8) (p.11), a semiempirical model was proposed for $\dot{\phi}_i$ resulting from a drive of constant amplitude, F_D, and rise time, T_r. According to this model,

$$\dot{\phi}_i = \rho_i (F_D - F_d^n) e^{-(t-T_r)(F_D - F_d^n)/C_i} , \qquad (4)$$

where ρ_i is switching resistance per turn squared, F_d^n is threshold, and C_i is a constant of proportionality of the decay time constant. Equation (4) was based [*cf.* Report 3, pp. 11-12] on the hypothesis that $\dot{\phi}_i$ results from the motions of those domain walls that do not collide with each other, at least in the beginning of switching, each terminating when the wall is obstructed by an inhomogeneity (energy hill of a high slope). The decaying waveform of $\dot{\phi}_i$ stems from the random distributions of the distance to the obstructing hill and the average slope of nonobstructing hills. The higher is the excess of F over a threshold F_d^n, the faster the motions of these walls are. Hence, $\dot{\phi}_i$ is proportional and the decay time constant is inversely proportional to $(F - F_d^n)$.

b. LIMITATIONS OF THE PREVIOUS MODEL

The limitations of the model for $\dot{\phi}_i$ in Report 3 (pp. 11-12) are as follows:

(1) The model does not account for the rise of $\dot{\phi}_i$ from zero to its peak value at $t = T_r$ (while F rises from F_i to, say, 95 percent of F_D). In evaluating the sum $\dot{\phi}_\epsilon + \dot{\phi}_i$ in Fig. 4 of Report 3, $\dot{\phi}_i$ is neglected during $t < T_r$, whereas $\dot{\phi}_\epsilon$ is neglected from $t = T_r$ until $F = F_D$.

(2) The model is not valid if the rise time, T_r, is relatively long.

(3) The model is applicable only if the drive mmf, F, is of constant amplitude, F_D.

(4) Further investigation of a given ferrite material has shown that if F_D/F_d^n is not much larger than unity, then $\dot{\phi}_i$ is proportional not to $F_D - F_d^n$ but rather to $(F_D - F_i^n)^{\nu_i}$, where $F_i < F_d^n$ and $\nu_i > 1$. This behavior is similar to that of the main $\dot{\phi}$, and may be attributed to the increase in the number of nucleation centers with the excess mmf.[6]

c. MODIFIED MODEL

Let T_i be the time at which F reaches the value F_i. The above-mentioned drawbacks are overcome if the model of Eq. (4) is modified to the following expression (which is valid if $F \geq F_i$ and thus for $t \geq T_i$):

$$\dot{\phi}_i = \lambda_i (F - F_i)^{\nu_i} e^{-(t-T_i)(F-F_i)/C_i} \quad . \tag{5}$$

Here, λ_i is a constant of proportionality [replacing ρ_i in Eq. (4), in analogy with the expressions $\lambda(F - F_0'')^{\nu}$ and $\rho_p(F - F_0)$ for $\dot{\phi}_p$ in the model for the main $\dot{\phi}$, Report 3, Eq. (2)] and F has an arbitrary waveform with an arbitrary rise time, T_r. From $t = 0$ to $t = T_i$, $\dot{\phi}_i = 0$. During $T_i \leq t \leq T_r$, $\dot{\phi}_i(t)$ increases independently of $\dot{\phi}_\epsilon$ to a peak value, $\dot{\phi}_{ip}$, at $t = T_{ip}$. If the average slope of the rise of $F(t)$ is low enough, then $T_{ip} < T_r$; otherwise, $T_{ip} = T_r$.

d. EXAMPLE

As an example, consider $F(t)$ with a ramp rise followed by a constant amplitude, *i.e.*,

$$F = \begin{cases} F_D t/T_r & \text{if } 0 \leq t \leq T_r \\ F_D & \text{if } T_r \leq t \end{cases} \quad . \tag{6}$$

Switching starts at

$$T_i = T_r \frac{F_i}{F_D} \quad . \tag{7}$$

During $T_i \leq t \leq T_r$, Eqs. (5), (6), and (7) give

$$\dot{\phi}_i = \lambda_i [(t - T_i)F_D/T_r]^{\nu_i} e^{-(t-T_i)^2 F_D/(C_i T_r)} \quad ; \tag{8a}$$

during $T_r \leq t$, Eqs. (5) and (6) give

$$\dot{\phi}_i = \lambda_i (F_D - F_i)^{\nu_i} e^{-(t-T_i)(F_D-F_i)/C_i} \quad . \tag{8b}$$

Whereas $\dot{\phi}_i(t)$ of Eq. [8(b)] falls exponentially with time, $\dot{\phi}_i(t)$ of Eq. [8(a)] rises with time to a peak value $\dot{\phi}_{ip}$ at $t = T_{ip}$. Differentiating Eq. [8(a)] with respect to time and equating $d\dot{\phi}_i/dt$ to zero, we get

$$T_{ip} = F_i(T_r/F_D) + \sqrt{(C_i\nu_i/2) \cdot (T_r/F_D)} \quad , \tag{9}$$

which is valid provided that $T_{ip} \leq T_r$. For a given value of F_D, $T_{ip} < T_r$ if T_r exceeds a certain value, T_{rB}. By equating T_{ip}, Eq. (9), to T_r, we find that

$$T_{rB} = \frac{C_i\nu_i}{2F_D[1 - (F_i/F_D)]^2} \quad . \tag{10}$$

On the other hand, if $T_r > T_{rB}$, then the expression given by Eq. (9) is larger than T_r. Since for $t > T_r$, $\dot{\phi}_i(t)$ in Eq. [(8(b)] decays exponentially, $\dot{\phi}_i(t)$ reaches a discontinuous peak at $t = T_r$, and so $T_{ip} = T_r$.

We may thus conclude that, for a given F_D value, if $T_r \geq T_{rB}$, Eq. (10), then $T_{ip} \leq T_r$ and Eq. (9) is valid; but if $T_r \leq T_{rB}$, then $T_{ip} = T_r$. The two cases, designated by Subscripts (1) and (2), are shown in Fig. 4. In both cases $\dot{\phi}(t)$ is discontinuous at $t = T_r$, but in Case (2) the discontinuity and the peak of $\dot{\phi}_i(t)$ coincide. In each case, the peak value of $\dot{\phi}_i$, $\dot{\phi}_{ip}$, may be determined by substituting $t = T_{ip}$ into Eq. [8(a)]. Thus,

$$\dot{\phi}_{ip} = \begin{cases} \lambda_i\left(\dfrac{C_i\nu_i F_D}{2eT_r}\right)^{\nu_i/2} & \text{if } T_r \geq T_{rB} \quad \text{[Case (1)]} \\ \lambda_i(F_D - F_i)^{\nu_i} e^{-T_r(F_D-F_i)^2/(F_D C_i)} & \text{if } T_r \leq T_{rB} \quad \text{[Case (2)]} \end{cases} \tag{11}$$

We conclude from Eqs. (7) through (11) that, for a given value of F_D, the larger T_r is, the larger are T_i and T_{ip} and the smaller is $\dot{\phi}_{ip}$. In the limit of Case (1), as $T_r \to \infty$, $\dot{\phi}_i \to 0$. In the limit of Case (2), as $T_r \to 0$, $\dot{\phi}_{ip} \to \lambda_i(F_D - F_i)^{\nu_i}$; hence, $\dot{\phi}_i$ rises instantaneously to $\dot{\phi}_{ip}$ at $t = 0$ and decays exponentially thereafter in accordance with Eq. [8(b)] in which $T_i = 0$.

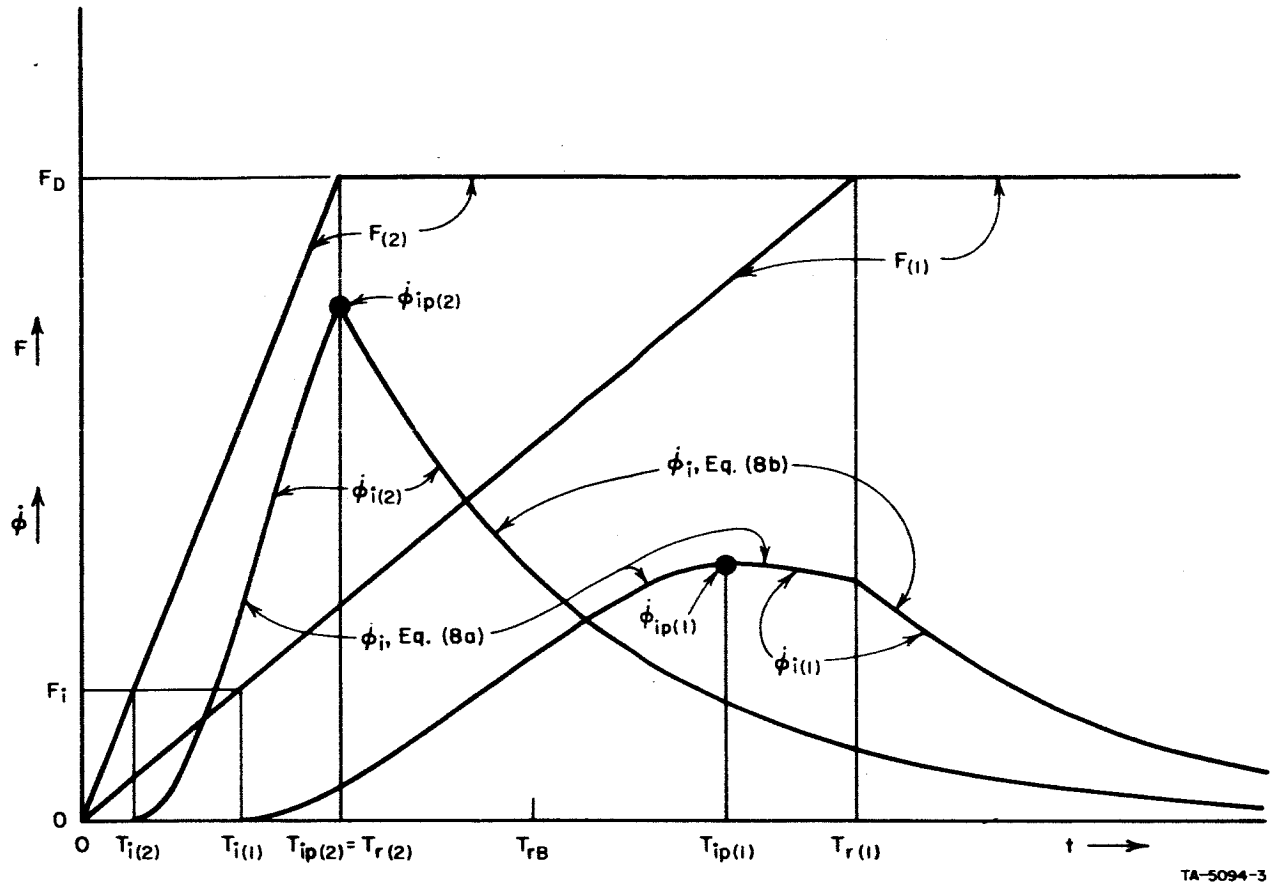


FIG. 4 EFFECT OF T_r ON $\dot{\phi}_i(t)$ WAVEFORM FOR A GIVEN F_D

A similar analysis may be carried out for a given value of T_r and a variable F_D. From Eq. (9) we find the value

$$F_{DB} = F_i + \frac{C_i \nu_i}{4T_r} + \sqrt{\frac{C_i \nu_i}{4T_r}\left(2F_i + \frac{C_i \nu_i}{4T_r}\right)} \qquad (12)$$

such that if $F_D \geq F_{DB}$, then $T_{ip} \leq T_r$, but if $F_D \leq F_{DB}$, then $T_{ip} = T_r$.

e. EFFECTS OF GEOMETRY ON SWITCHING PARAMETERS

In analogy with the material inelastic-switching properties of the main $\dot{\phi}$ of a thin core (or leg) of cross-sectional area A and average length, l (*cf.* Report 2, pp. 8, 37-40), Eq. (5) may be converted

from $\dot{\phi}_i(F, t)$ into $\dot{B}_i(H, t)$, where $\dot{B}_i = \dot{\phi}_i/A$ and $H = F/l$. Thus, in analogy with the relation $B_p = \kappa(H - H_0'')^{\nu}$ of Eq. (83) in Report 2,

$$\dot{B}_i = \kappa_i (H - H_i)^{\nu_i} e^{-(t-T_i)(H-H_i)/M_i} \quad , \tag{13}$$

where κ_i, H_i, and M_i are material parameters (H reaches the threshold value H_i at $t = T_i$). Following the geometrical relations expressed by Eqs. (91) and (92) of Report 2 (p. 40),

$$\lambda_i = \kappa_i A / l^{\nu_i} \tag{14}$$

and

$$F_i = H_i l \quad . \tag{15}$$

Assuming that the decay time constant is determined solely by the material-switching behavior,

$$C_i = M_i l \quad . \tag{16}$$

Equations (14) through (16) are useful in calculation of $\dot{\phi}_i$ of legs (or cores) of the same material, but of different geometry, *e.g.*, in calculation of flux division in a saturable three-leg core, as we shall see later.

f. VARIATION OF F_i

We have presumed that $\dot{\phi}_i$ is generated by the motion of noncolliding walls which are finally obstructed by randomly distributed centers of imperfections (energy hills of high slopes). In line with our hypothesis that the average slope of the nonobstructing hills is randomly distributed, the slope of the first hill is also randomly distributed. As a result, we may expect the threshold F_i to increase from near zero to some finite asympotic value, F_{0i}. Physically, this means that a small percentage of the walls are initially at a barely stable state (*i.e.*, a very small applied F will displace them inelastically) and that most of the walls will break free if $F \geq F_{0i}$.

A plot of F_i *vs.* F is shown in Fig. 5. As an approximation, F_i *vs.* F may be described by a tanh function, *i.e.*,

$$F_i = F_{0i} \tanh (F/F_{0i}) \quad . \tag{17}$$

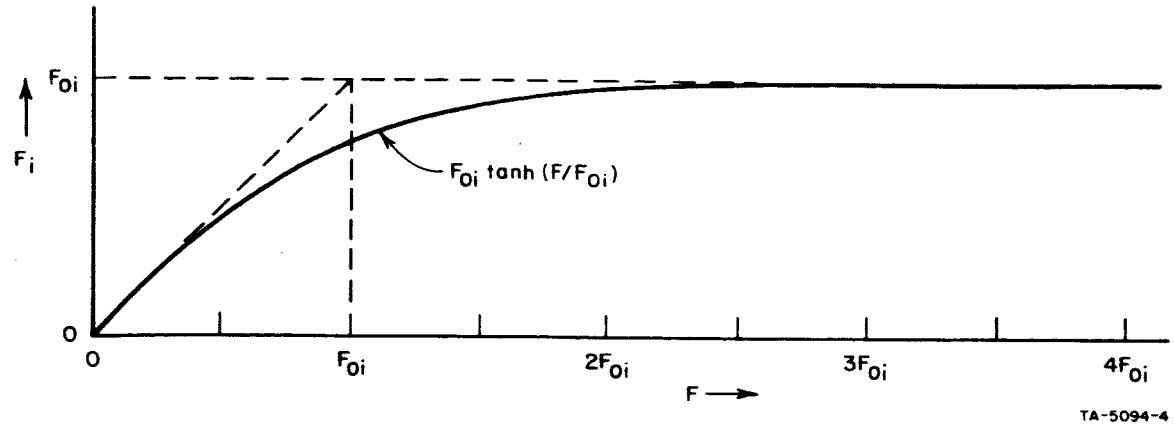


FIG. 5 APPROXIMATE VARIATION OF F_i WITH F

Note that the slope of $F_i(F)$ at $F = 0$ is unity, as expected from the discussion above. Thus, for F values of F_{0i}, $2F_{0i}$, and $3F_{0i}$, F_i reaches 76.16 percent, 96.4 percent and 99.5 percent of F_{0i}, respectively. With F_i defined as in Eq. (17), $T_i = 0$ because $F = F_i$ at $t = 0$.

The relation between the value of F_{0i} and the values of other threshold parameters will be discussed later (p. 36).

3. Inelastic Main $\dot{\phi}$

a. INTRODUCTION

A model for the main $\dot{\phi}$ was proposed and applied in Report 3. Following Eqs. (1), (2) and (4) of Report 3,

$$\dot{\phi}_{ma} = \dot{\phi}_p \eta \quad , \tag{18}$$

where

$$\dot{\phi}_p = \begin{cases} 0 & \text{if } F \leq F_0'' \\ \lambda(F - F_0'')^{\nu} & \text{if } F_0'' \leq F \leq F_B \\ \rho_p(F - F_0) & \text{if } F_B \leq F \end{cases} \tag{19}$$

and

$$\eta = 1 - \left(\frac{2\phi + \phi_s - \phi_d}{\phi_s + \phi_d} \right)^2 \tag{20}$$

and where ϕ_d is the ϕ value on the static $\phi(F)$ curve. Based on experimental verification in Report 3, we concluded [*cf.* Report 3, p. 33] that the above model is satisfactory, except if F is low, *i.e.*, around the coercive mmf or lower.

b. MODIFIED $\eta(\phi)$

Typical $\dot{\phi}(\phi)$ oscillogram of step-F switching in a thin ferrite core (Core E-6, Report 3, p. 23) are shown in Fig. 6 for three amplitude values: F_D = 1.2, 1.4, and 1.8 amp-turns. In the beginning of switching, as ϕ rises by a small $\Delta\phi$ above $-\phi_r$, $\dot{\phi}(\phi)$ is due primarily to $\dot{\phi}_\epsilon$ and $\dot{\phi}_i$. Beyond this region of ϕ, $\dot{\phi}(\phi)$ is due primarily to $\dot{\phi}_i$ and $\dot{\phi}_{ma}$. As ϕ increases, the contribution of $\dot{\phi}_i$ to the total $\dot{\phi}$ diminishes to a negligible amount, and $\dot{\phi}(\phi)$ is essentially $\dot{\phi}_{ma}(\phi)$. An extrapolation of $\dot{\phi}_{ma}(\phi)$ to the ϕ axis is dash-lined in each case of Fig. 6. Each $\dot{\phi}_{ma}(\phi)$ extrapolation intersects the ϕ axis at $\phi \cong -\phi_r$ rather than $\phi \cong -\phi_s$. As a result of this observation, we shall modify $\eta(\phi)$ of Eq. (20) by replacing ϕ_s by ϕ_r, *i.e.*,

$$\eta = 1 - \left(\frac{2\phi + \phi_r - \phi_d}{\phi_r + \phi_d}\right)^2 \quad . \tag{21}$$

Note that ϕ is the total flux, *i.e.*, obtained by integration of the three components of $\dot{\phi}$, Eq. (1).

4. DISCUSSION

The experimental data given in Fig. 6 were known before the previous $\eta(\phi)$ function, Eq. (20), was proposed. One may then ask why we chose $\eta(\phi)$ given by Eq. (20) instead of $\eta(\phi)$ given by Eq. (21). The answer lies in the solution to the differential equation of $\dot{\phi}_{ma}$ in the absence of the decaying component, $\dot{\phi}_i$. In order to simplify the explanation of this point, let us assume an ideal step-F switching (F rises to F_D in a zero rise time), for which $\dot{\phi}_p$, Eq. (19), is constant. We shall first neglect the elastic $\dot{\phi}$ component, $\dot{\phi}_\epsilon$. If ϕ is identified with the main component, ϕ_{ma} (*i.e.*, the flux due to time integration of $\dot{\phi}_{ma}$ only), then the resulting solution of the differential equation expressed by Eq. (18) in which $\eta(\phi)$ is given by Eq. (21) is absurd: The initial value of $\dot{\phi}$ is zero and the switching time required to change ϕ from $-\phi_r$ to zero is infinite [*cf.* Report 1, Eq. (40) and Fig. 30, pp. 27-28]. In order to overcome this difficulty,

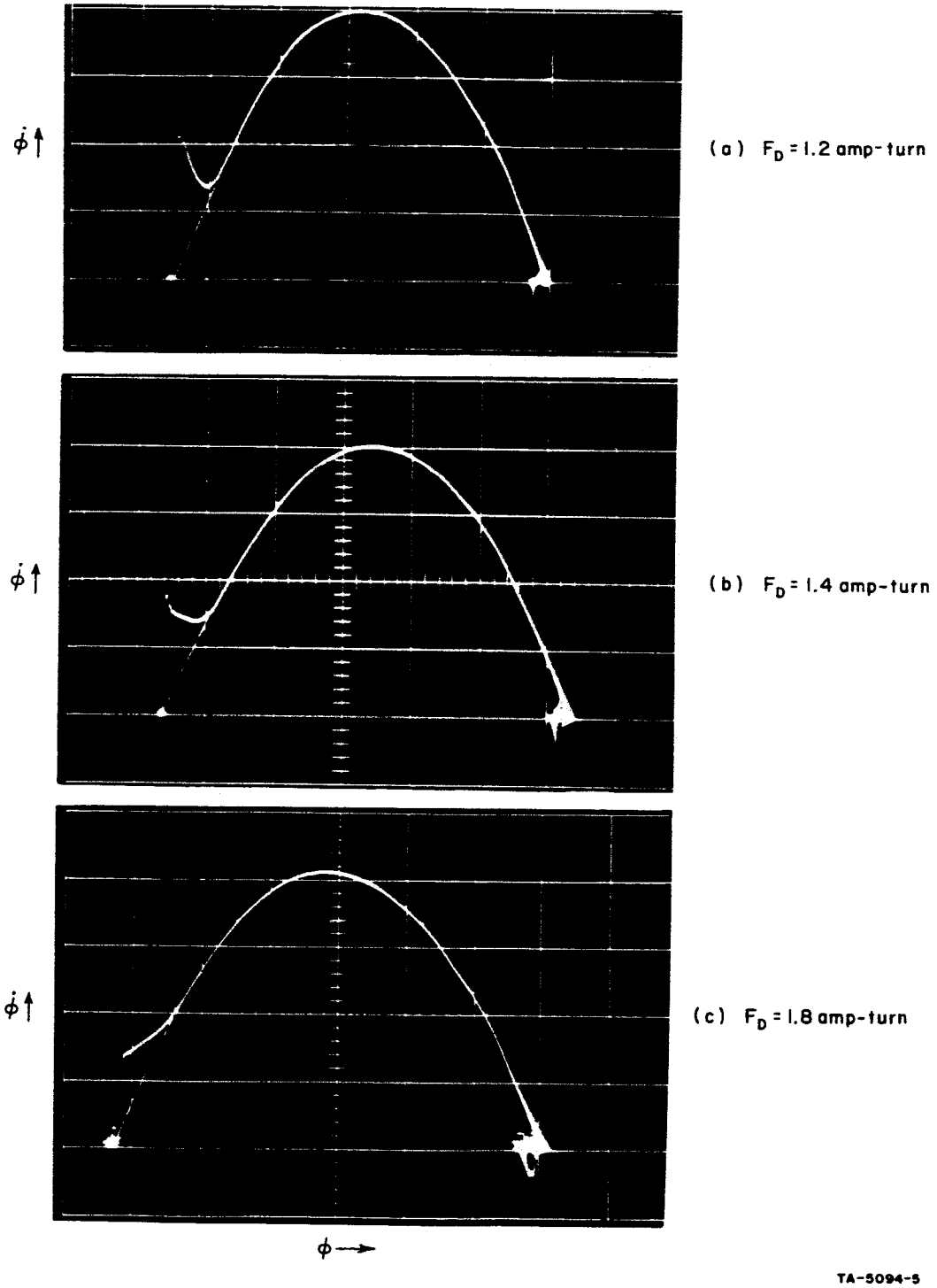


FIG. 6 $\dot{\phi}(\phi)$ OSCILLOGRAMS OF STEP-F SWITCHING OF A THIN FERRITE RING
Drive: 0.05 μsec rise time; variable amplitude F_D.
Core: E-6; OD/ID = 1.06; F_c = 0.9 amp-turn.
Added dashed lines are extrapolated $\dot{\phi}_{ma}$ vs. ϕ.
ϕ Scale = 1.04 maxwell/major div.; $\dot{\phi}$ scale: (a) 3.3 mv/turn, (b) 6.9 mv/turn, (c) 13.8 mv/turn.

we *approximated* $\eta(\phi)$ by the expression given in Eq. (20). However, this difficulty in obtaining the proper rate of increase of ϕ from $-\phi_r$ does not exist with our present model because of the presence of the additional, decaying, $\dot{\phi}$ component, whose initial value is finite, *i.e.*, $\lambda_i(F_D - F_i)^{\nu_i}$. Under this condition, we are allowed to use the more exact $\eta(\phi)$ function given by Eq. (21).

In the discussion above we have neglected the elastic component, $\dot{\phi}_\epsilon$. If we include $\dot{\phi}_\epsilon$, then, in the absence of $\dot{\phi}_i$, $\dot{\phi} = \dot{\phi}_{ma} + \dot{\phi}_\epsilon$, and as F rises from zero to F_D, ϕ increases by the amount ϵF_D. Although the use of $\eta(\phi)$ of Eq. (21) will not lead to an infinite switching time, the initial value of $\dot{\phi}_{ma}$ at $\phi = -\phi_r + \epsilon F_D$ is extremely low, and the resulting computed $\dot{\phi}$ waveform will have little resemblance to the observed $\dot{\phi}$ waveform. Therefore, even if $\dot{\phi}_\epsilon$ is included but $\dot{\phi}_i$ is not included in the total $\dot{\phi}$, we have to use the $\eta(\phi)$ function given in Eq. (20) and not the one given in Eq. (21).

If F_D is larger than the F value at the upper knee of the static $\phi(F)$ curve, then the sum $\dot{\phi}_i + \dot{\phi}_{ma}$, in which $\dot{\phi}_{ma}$ is calculated by using $\eta(\phi)$ of Eq. (21), is approximated quite well by $\dot{\phi}_{ma}$ calculated by using $\eta(\phi)$ of Eq. (20), as is illustrated in Fig. 7. The resemblance between the two $\dot{\phi}(t)$ waveforms in Figs. 7(a) and 7(b) justifies the practice of using the approximation

$$\dot{\phi}_p\{1 - [(2\phi + \phi_s - \phi_d)/(\phi_s + \phi_d)]^2\} \cong \dot{\phi}_i + \dot{\phi}_p\{1 - [(2\phi + \phi_r - \phi_d)/(\phi_r + \phi_d)]^2\} \tag{22}$$

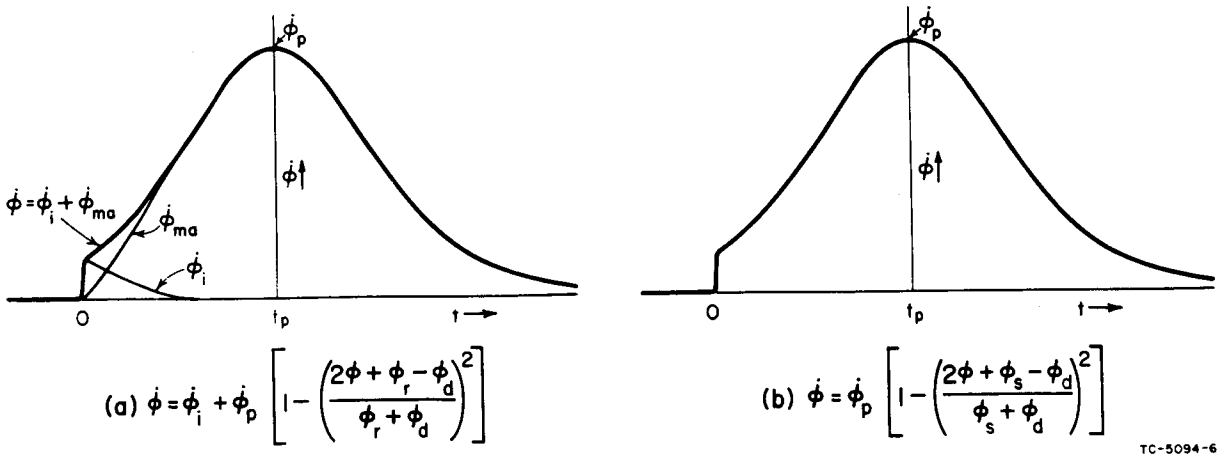


FIG. 7 RESEMBLANCE BETWEEN $\dot{\phi}(t)$ WAVEFORMS DERIVED FROM TWO MODELS FOR INELASTIC $\dot{\phi}$

if the drive mmf is large enough to switch ϕ to the saturation region above the knee. However, if F is low (*e.g.*, $F \cong F_c$), then such an approximation may become quite poor. Furthermore, if $F_i < F < F_0''$, then $\dot{\phi}_i > 0$, whereas $\dot{\phi}_{ma} = 0$ regardless of which $\eta(\phi)$ function is used because, following Eq. (19), $\dot{\phi}_p = 0$.

In general, if the instantaneous $\dot{\phi}$ is of interest, the elastic component of $\dot{\phi}$, $\dot{\phi}_\epsilon$, should be included. However, if inelastic $\Delta\phi$ is to be calculated, then it is justified to neglect $\dot{\phi}_\epsilon$.

5. Summary

Three components are distinguished in the total $\dot{\phi}(t)$ waveform of a ferrite core: elastic $\dot{\phi}$ spike, $\dot{\phi}_\epsilon$; inelastic decaying $\dot{\phi}$, $\dot{\phi}_i$; and the bell-shaped main inelastic $\dot{\phi}$, $\dot{\phi}_{ma}$. The following semiempirical switching models are proposed for the three components of $\dot{\phi}$: For the elastic $\dot{\phi}$, $\dot{\phi}_\epsilon = \epsilon \dot{F}$, where ϵ is a function of F, ϕ, and the previous switching; for $\phi = -\phi_r$ and F not much larger than F_c,

$$\epsilon = \{(\phi_s - \phi_r)/[2\pi(r_o - r_i)H_a]\} \ln(r_o/r_i) \quad .$$

For the decaying inelastic $\dot{\phi}$, $\dot{\phi}_i = \lambda_i (F - F_i)^{\nu_i} e^{-(t-T_i)(F-F_i)/C_i}$, where λ_i is a proportionality factor, F_i is the threshold, ν_i is a power coefficient, T_i is the time of beginning of $\dot{\phi}_i$ switching (when F reaches F_i), and C_i is a constant proportional to the decay time constant. For the main inelastic $\dot{\phi}$, $\dot{\phi}_{ma} = \dot{\phi}_p\{1 - [(2\phi + \phi_r - \phi_d)/(\phi_r + \phi_d)]^2\}$, where, for given F, $\dot{\phi}_p$ is the peak value of $\dot{\phi}$, ϕ_d is the ϕ value on the static $\phi(F)$ curve and ϕ_r is the maximum residual flux. If F is low (around F_c or lower), $\dot{\phi}_i$ should be distinguished from $\dot{\phi}_{ma}$, but if F is large, then $\dot{\phi}_i + \dot{\phi}_{ma}$ may be approximated quite well by

$$\dot{\phi} = \dot{\phi}_p\{1 - [(2\phi + \phi_s - \phi_d)/(\phi_s + \phi_d)]^2\}$$

in which ϕ_s is saturation flux.

C. COMPUTER PROGRAMS FOR $\dot{\phi}$ COMPONENTS

In Report 3, a computer program for only the main component of $\dot{\phi}$ was provided. It was felt then (*cf.* Report 3, p. 34) that more needed to be known about $\dot{\phi}_\epsilon$ and $\dot{\phi}_i$ before these components were incorporated into the

over-all computer program. After studying $\dot{\phi}_i$ in more detail, its incorporation has been undertaken. (It should be emphasized that our present information is based on the study of one core only; more core materials will be investigated in the future.)

If a core is switched unloaded, then $F(t)$ and $\dot{F}(t)$ are given. However, if a loaded core is switched, then the time variables, including F and $\dot{F}$, are solved for transcendentally. If Newton's method (*cf.* Report 3, p. 42) is used in this solution, then the value of $\dot{\phi}' = d\dot{\phi}/dF$ needs to be known. Following Eq. (1),

$$\dot{\phi}' = \dot{\phi}'_\epsilon + \dot{\phi}'_i + \dot{\phi}'_{ma} \quad , \tag{23}$$

where $\dot{\phi}'_\epsilon = d\dot{\phi}_\epsilon/dF$, $\dot{\phi}'_i = d\dot{\phi}_i/dF$, and $\dot{\phi}'_{ma} = d\dot{\phi}_{ma}/dF$. Each of these $\dot{\phi}'$ components will be computed in the corresponding $\dot{\phi}$ PROCEDURE where the $\dot{\phi}$ component itself is being computed.

1. Computation of $\dot{\phi}_\epsilon$ and $\dot{\phi}'_\epsilon$

Computation of $\dot{\phi}_\epsilon$ is based on Eq. (2), *i.e.*, $\dot{\phi}_\epsilon = \epsilon\dot{F}$. A computer program for computing $\dot{\phi}_\epsilon$ and $\dot{\phi}'_\epsilon$ is given in Appendix A in a form of PROCEDURE $\dot{\phi}_\epsilon(\dot{F}, \Delta t, NV, \dot{\phi}'_\epsilon)$. The elastic switching parameter, ϵ, is global, *i.e.*, declared throughout the program; it may be evaluated by using Eq. (3). The input parameters are $\dot{F}$, Δt, and NV, the latter standing for a negligible value of $\dot{\phi}_\epsilon$; $\dot{\phi}'_\epsilon$ on the other hand, is an output parameter.

In the case of a loaded core, $\dot{F}$ at $t = t_n = n\Delta t$ is approximated by differences rather than differentials, *i.e.*,

$$\dot{F} \cong \frac{F_n - F_{n-1}}{\Delta t} \quad . \tag{24}$$

Since only F_n is solved for (the values of F_{n-1} and Δt are given), $\dot{\phi}'_\epsilon = d(\epsilon\dot{F})/dF_n$. Hence,

$$\dot{\phi}'_\epsilon = \epsilon/\Delta t \quad . \tag{25}$$

For practical consideration, if $\dot{\phi}_\epsilon < NV$, we shall assume that $\dot{\phi}_\epsilon = 0$ and, therefore, also $\dot{\phi}'_\epsilon = 0$.

2. Computation of $\dot{\phi}_i$ and $\dot{\phi}'_i$

Based on Eq. (5), $\dot{\phi}_i$ is computed as a function of F, t, and T_i. A computer program for computing $\dot{\phi}_i$ and $\dot{\phi}'_i = d\dot{\phi}_i/dF$ is given in Appendix B in the form of PROCEDURE $\dot{\phi}_i(F, t, T_i, \dot{\phi}'_i)$. The core parameters λ_i, F_i, ν_i and C_i are assumed to be global parameters. The excess mmf, $F - F_i$, is computed each time $\dot{\phi}_i(F, t, T_i, \dot{\phi}'_i)$ PROCEDURE is called, and as soon as $F - F_i > 0$, T_i is identified with the corresponding t value. Using Eq. (5),

$$\dot{\phi}_i = \begin{cases} 0 & \text{if } t < T_i \\ \lambda_i (F - F_i)^{\nu_i} e^{-(t-T_i)(F-F_i)/C_i} & \text{if } t \geq T_i \end{cases} . \tag{26}$$

Differentiation of Eq. (26) with respect to F gives

$$\dot{\phi}'_i = \begin{cases} 0 & \text{if } t < T_i \\ \dot{\phi}_i \left(\dfrac{1}{F - F_i} - \dfrac{t - T_i}{C_i} \right) & \text{if } t \geq T_i \end{cases} . \tag{27}$$

For practical consideration, if $\dot{\phi}_i$ is negligible, *e.g.*, $\dot{\phi}_i < 0.001\, \lambda_i (F - F_i)$, then we assume that $\dot{\phi}_i = 0$ and $\dot{\phi}'_i = 0$

3. Computation of $\dot{\phi}_{ma}$ and $\dot{\phi}'_{ma}$

Computation of the main component of $\dot{\phi}$, $\dot{\phi}_{ma}$ is based on Eqs. (18), (19), and (21). The computer program for computing $\dot{\phi}_{ma}$ and $\dot{\phi}'_{ma} = d\dot{\phi}_{ma}/dF$ is given in Appendix C in the form of PROCEDURE $\dot{\phi}_{ma}(F, \phi, \phi_d, \dot{\phi}'_{ma})$, in which F and ϕ are input parameters and ϕ_d and $\dot{\phi}'_{ma}$ are output parameters. This PROCEDURE is identical with $\dot{\phi}(F, \phi, \phi_d, \dot{\phi}')$ PROCEDURE given in Report 3 (Appendix A, pp. 133-135) except for two modifications: First, $\eta(\phi)$ follows Eq. (21) instead of Eq. (20) and second, F_{12}, F_{23}, V_1, and V_2 [*cf.* Report 3, Eqs. (26) through (29), pp. 18-19] are treated as global core parameters instead of being computed once inside the $\dot{\phi}$ PROCEDURE. Following the first modification, ϕ_r replaces ϕ_s in evaluation of $\dot{\phi}'_{ma}$ [*cf.* Report 3, Eq. (41), p. 20]. For practical consideration, if $\phi_d - \phi \leq 0.001\phi_r$, then we assume that $\dot{\phi}_{ma} = 0$ and therefore, also $\dot{\phi}'_{ma} = 0$.

We have shown [*cf.* Eq. (22)] that if F is not low, $(\dot{\phi}_i + \dot{\phi}_{ma})$ may be replaced by $\dot{\phi}_{ma}$ alone if we replace $\eta(\phi)$ of Eq. (21) by $\eta(\phi)$ of Eq. (20). Under this condition, PHIS should replace PHIR in Lines PHDTMA42 through PHDTMA45 of the $\dot{\phi}_{ma}(F, \phi, \phi_d, \dot{\phi}'_{ma})$ PROCEDURE, Appendix C. In Appendix D, two computer PROCEDUREs of different output parameters are given for this case: $\dot{\phi}(F, \phi, \phi_d, \dot{\phi}')$ and $\dot{\phi}(F, \phi, \dot{\phi}', \dot{\phi}^*)$. The first PROCEDURE is the same as the one used in Appendix A of Report 3 (p. 133), and its output furnishes the values of $\dot{\phi}$, ϕ_d, and $\dot{\phi}'$. The output of the second PROCEDURE includes $\dot{\phi}$, $\dot{\phi}'$ and $\dot{\phi}^*$, where, as shown on p. 43 of Report 3,

$$\dot{\phi}^* = \frac{\partial \dot{\phi}}{\partial \phi} = -4\dot{\phi}_p \frac{2\phi + \phi_s - \phi_d}{(\phi_s + \phi_d)^2} \quad . \tag{28}$$

Another difference, which is minor, is that the core parameters F_{12}, F_{23}, V_1 and V_2 are arbitrarily treated as global parameters in the $\dot{\phi}(F, \phi, \dot{\phi}', \dot{\phi}^*)$ PROCEDURE, but not in the $\dot{\phi}(F, \phi, \phi_d, \dot{\phi}')$ PROCEDURE.

4. Summary

Computer programs for the three components of $\dot{\phi}$ and $\dot{\phi}' = d\dot{\phi}'/dF$ are given in PROCEDURE forms in Appendices A, B, and C. PROCEDURE $\dot{\phi}_\epsilon(F, \Delta t, NV, \dot{\phi}'_\epsilon)$ is based on Eqs. (2) and (25); PROCEDURE $\dot{\phi}_i(F, t, T_i, \dot{\phi}'_i)$ is based on Eqs. (26) and (27); and PROCEDURE $\dot{\phi}_{ma}(F, \phi, \phi_d, \dot{\phi}'_{ma})$ is based on Eqs. (18), (19), and (21). Two computer PROCEDUREs that differ in their output parameters, $\dot{\phi}(F, \phi, \phi_d, \dot{\phi}')$ and $\dot{\phi}(F, \phi, \dot{\phi}', \dot{\phi}^*)$, are given in Appendix D for the case in which the sum $\dot{\phi}_i + \dot{\phi}_{ma}$ is approximated by $\dot{\phi} = \dot{\phi}_p\{1 - [(2\phi + \phi_s - \phi_d)/(\phi_s + \phi_d)]^2\}$, Eqs. (18) through (20).

D. EXPERIMENTAL VERIFICATION

The experimental study of the $\dot{\phi}$ components consisted of clearing and setting a thin ferrite core (Core E-6, Report 3, p. 23), and photographing the waveforms of $\dot{\phi}(t)$ and $F(t)$ during the beginning of the SET pulse. Variations in the SET pulse included two rise-time values, each with seven or eight different amplitude values. A computer program was written and, with the proper core and circuit parameters, used to compute the $\dot{\phi}(t)$ waveforms. Experimental and computed results were then compared and analyzed.

1. Experiment

a. Introduction

In order to investigate the decaying component, $\dot{\phi}_i$, meaningfully, the main component, $\dot{\phi}_{ma}$, should be as small as possible, certainly not much larger than $\dot{\phi}_i$. This condition can be achieved if the rise time of $F(t)$ is short. On the other hand, a short rise time generates a high-frequency ringing in the core windings (due to stray capacitance), causing a distortion in $\dot{\phi}(t)$. Another difficulty in this type of investigation stems from the variations of $\dot{\phi}_\epsilon$ if the rising portion of $F(t)$ is not smooth.

b. Rise Time

Two values of rise time were used: one was around 0.1 μsec, and the other around 0.02 μsec. No difficulties were encountered using the longer rise time. However, some ringing was present in the case of $T_r \tilde{=} 0.02$ μsec. A typical example is shown in Fig. 8, where the waveform of the observed $\dot{\phi}(t)$ has been traced. It can be seen that subtraction of a decaying $\dot{\phi}(t)$ ringing of a high frequency (about 100 megacycles) from the observed $\dot{\phi}(t)$ oscillogram results in a smoother $\dot{\phi}(t)$ waveform. The latter should be considered to be the actual $\dot{\phi}(t)$ waveform.

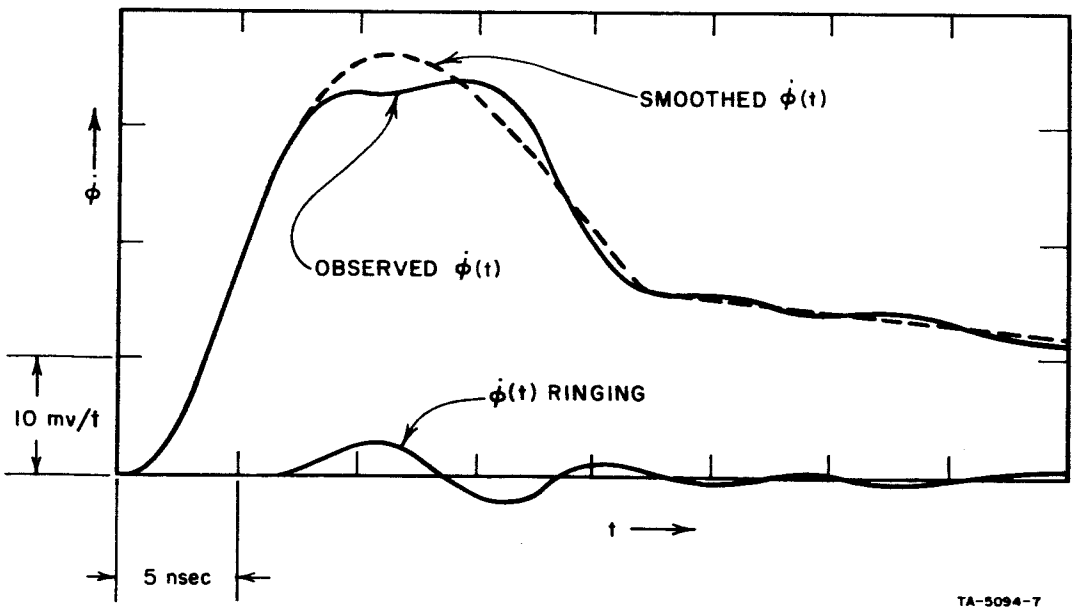


FIG. 8 RINGING IN $\dot{\phi}(t)$ FOR F(t) WITH A SHORT RISE TIME
F_D = 1.50 amp-turn; T_r = 19 nanoseconds.

A mercury-relay pulser, with its capability of providing current pulses up to 40-ampere amplitude with less than 0.7-nanosecond rise time, is very attractive for studying high-F $\dot{\phi}_\epsilon$ and $\dot{\phi}_i$. Unfortunately, an attempt to use the mercury relay has been unsuccessful so far because of an excessive ringing in the $\dot{\phi}(t)$ waveforms. This problem should be pursued further in the future.

c. TESTED CORE

Oscillograms of $\dot{\phi}(t)$ were recorded only for one core, the thin ferrite toroid (OD/ID = 1.06) which is referred to as Core E-6 in Report 3. The dimensions and switching parameters of this core may be found in Report 3, p. 23. The nominal composition of the core material (commercially known as Telemeter Magnetics T-5) is

$$[Mg_{0.32}Zn_{0.10}Mn_{0.58}]^{++}[Mn_{0.26}Fe_{0.74}]_2^{+++}O_4 \quad .$$

The $\dot{\phi}(t)$ waveforms of additional cores of different material and larger OD/ID ratio were observed and appeared to behave in a manner similar to that of the $\dot{\phi}(t)$ of Core E-6.

d. CORE HOLDER

The same coaxial core holder was used as described on p. 85 of Report 3. The sense winding was increased to 20 turns of No. 48 copper wire. The negative clear winding consisted of 10 turns distributed around the circumference of the core. The drive winding was modified so as to have a single turn made of six No. 48 copper wires. The pulses with 0.1-μsec rise time were applied to this single-turn 6-conductor winding. In this case the center conductor of the 50-ohm transmission line was not used. The pulses with 0.02-μsec rise time were applied via the center conductor of the 50-ohm line.

e. PULSE SEQUENCE

The pulse sequence was that discussed in Report 3, p. 83: first, a positive CLEAR pulse; second, a negative CLEAR pulse; and third, the (positive) SET pulse during which measurements were made.

f. EQUIPMENT

Four different current pulsers were used in this experiment:

(1) The SET pulses with 0.1-μsec rise time* were generated by paralleling several high-impedance transistor current drivers (Digital Equipment Corp., Model 62). Maximum amplitude was 5.0 amperes.

(2) The SET pulses with 0.02-μsec rise time* were generated by a tube current driver (Hewlett-Packard, Model 214A). Maximum amplitude was 2.0 amperes.

(3) The negative CLEAR pulse was generated by paralleling five tube drivers (Digital Equipment Corp., Model 50; 0.1-μsec rise time). Maximum amplitude was around 7 amperes.

(4) The positive CLEAR pulse (1.4 ampere × 10 turns) was generated by a single tube driver (Digital Equipment Corp., Model 51; 0.1-μsec rise time) for the cases of $T_r \cong 0.1$ μsec and by two transistor drivers (Digital Equipment Corp., Model 63; 0.05-μsec rise time) in parallel for the cases of $T_r \cong 0.02$ μsec.

Two oscilloscopes were used. The waveforms corresponding to 0.02-μsec rise time were recorded on a Hewlett-Packard 185A sampling oscilloscope, having a response time of about 0.5 nsec. The waveforms corresponding to 0.1-μsec rise time were recorded on a Tektronix 545 oscilloscope with a Type K plug-in unit, resulting in a combined response time of 13 nsec. The response of the Tektronix oscilloscope together with the plug-in unit was checked with a Tektronix Model 108 mercury pulser in order to make sure that no overshoot was obtained and that the response time was short enough.

The delay experienced by the $F(t)$ pulse with 0.02-μsec rise time (between the time when $\dot{\phi}$ was measured and the time when F was measured) was 3.5 nsec. This time delay was corrected by shifting the oscilloscope trace of $F(t)$ with relation to the $\dot{\phi}(t)$ trace and by photographing the two traces separately.

The temperature of the core was automatically maintained at 30°C by means of a thermistor probe and an electrical heater imbedded in the outer conductor of the coaxial core holder.

---

* The rise time T_r is twice the time that it takes the current pulse to reach half of its amplitude as defined later in Fig. 9.

2. COMPUTATION

a. METHOD OF COMPUTATION

Computation of $\dot{\phi}$ of an unloaded core may be performed by the same simple *predictor-corrector* method used in Report 3 [Eqs. (42), (43), and (44), p. 25], except that $\dot{\phi}$ now includes three components [$\dot{\phi}_\epsilon$, Eq. (2); $\dot{\phi}_i$, Eq. (5); and $\dot{\phi}_{ma}$, Eqs. (18), (19), and (21)] instead of one [$\dot{\phi}_{ma}$, Eqs. (18), (19), and (20)]. Following this method, ϕ_n at $t = t_n = n\Delta t$ (Δt is a short time interval compared with the switching time, τ_s) is first predicted from the relation

$$\phi_n = \phi_{n-2} + 2\Delta t\dot{\phi}_{n-1} \quad . \tag{29}$$

This is followed by an *iterative* computation of $\dot{\phi}_n$ and ϕ_n. Since $\dot{F}$ and F are both a function of time, $\dot{\phi}_n$ may be formally expressed as

$$\dot{\phi}_n = \dot{\phi}_\epsilon(t_n) + \dot{\phi}_i(t_n) + \dot{\phi}_{ma}(t_n, \phi_n) \quad . \tag{30}$$

The expression for the corrected ϕ_n is

$$\phi_n = \phi_{n-1} + 0.5\Delta t(\dot{\phi}_n + \dot{\phi}_{n-1}) \quad . \tag{31}$$

Equations (30) and (31) are used repeatedly until proper convergence of $\dot{\phi}_n$ and ϕ_n is achieved.

b. DRIVE CURRENT vs. TIME

As shown in Fig. 9, the waveform of the drive current, which is applied to the unloaded core, is approximated by the following functions:

$$i_D = \begin{cases} \dfrac{I_D}{2}\dfrac{t^2}{t_m^2}\left[\dfrac{t}{t_m}\left(ut_m - 2\right) - \left(ut_m - 3\right)\right] & \text{if } 0 \le t \le t_m \qquad (32a) \\ \dfrac{I_D}{2}\left[1 + \tanh u(t - t_m)\right] & \text{if } t_m \le t \quad , \qquad (32b) \end{cases}$$

where I_D is the amplitude, t_m is the "half rise time" (i.e., i_D reaches $I_D/2$ at $t = t_m$), and u is a waveform parameter (typically, ut_m varies

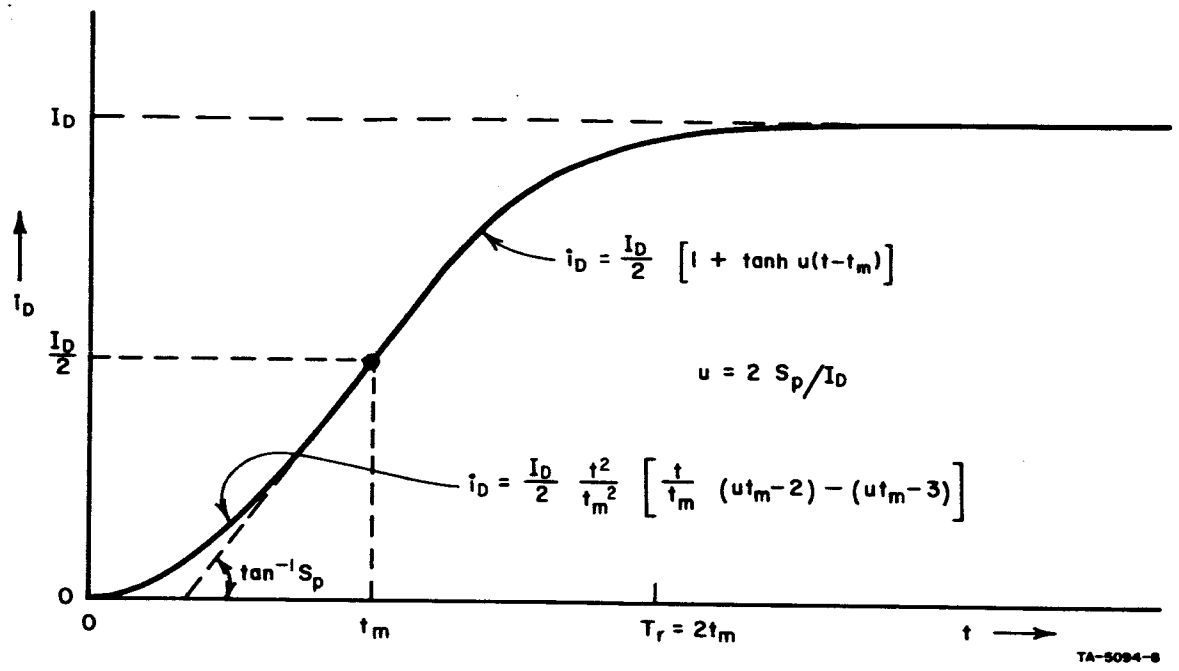


FIG. 9 APPROXIMATE DRIVE-CURRENT FUNCTION

from 1.35 to above 2). The functions given in Eqs. (32) satisfy the following requirements: $i_D = 0$ and $di_D/dt = 0$ at $t = 0$; the values of i_D and di_D/dt determined from Eq. (32a) are equal to the corresponding values determined from Eq. (32b) at $t = t_m$ (*i.e.*, no discontinuity at $t = t_m$); and $i_D \to I_D$ and $di_D/dt \to 0$ as $t \to \infty$. It can be seen by inspection of $i_D(t)$, Eqs. (32), and its time derivative,

$$\frac{di_D}{dt} = \begin{cases} \dfrac{I_D t}{2t_m^2}\left[\dfrac{t}{t_m}\,3(ut_m - 2) - 2(ut_m - 3)\right] & \text{if } 0 \le t \le t_m \qquad (33a) \\[2ex] \dfrac{I_D}{2}\,u\,\operatorname{sech}^2[u(t - t_m)] & \text{if } t_m \le t\ , \qquad (33b) \end{cases}$$

that these requirements are satisfied. (At $t = t_m$, $i_D = I_D/2$ and $di_D/dt = uI_D/2$.)

The maximum slope of $i_D(t)$, denoted by S_p, is reached at $t = t_m$. By measuring S_p, the value of u is readily determined from the relation.

$$u = 2S_p/I_D \quad . \tag{34}$$

As shown in Fig. 9, the rise time is defined as twice the "half rise time," *i.e.*, $T_r = 2t_m$.

c. OUTLINE FOR COMPUTER PROGRAM

A computer program for computing $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$ and the total $\dot{\phi}$ *vs.* time of an unloaded core is given in Appendix E. The outline of this program is as follows.

(1) Declare global identifiers of core parameters, circuit parameters, variables, miscellaneous, input-output lists and formats, and PROCEDUREs.

(2) Read in, compute, and print core and circuit parameters.

(3) Set the values of the switching parameters and the initial values of the variables.

(4) For every nth Δt during prescribed switching time T_s:

  (a) Compute $t_n = t_{n-1} + \Delta t$, i_{Dn} from Eq. (32), $F_n = N_D i_{Dn}$, F_{in} from Eq. (17), and F_n using Eq. (33).

  (b) Predict ϕ_n from Eq. (29), and compute the following variables in an iterative fashion (no more than six times) until the change in ϕ_n is negligible: $\dot{\phi}_{man}$ [call $\dot{\phi}_{ma}$ PROCEDURE], $\dot{\phi}_{\epsilon n}$ [call $\dot{\phi}_\epsilon$ PROCEDURE], $\dot{\phi}_{in}$ [call $\dot{\phi}_i$ PROCEDURE], $\dot{\phi}_n$ [Eq. (30)], and ϕ_n [Eq. (31)].

  (c) Reset index of variables before proceeding to the next Δt.

(5) Print output (t, i_D, $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$, $\dot{\phi}$, ϕ, ϕ_d, F, and number of iterations) every, say, second Δt during the rise of i_D and every tenth Δt thereafter.

d. CORE AND CIRCUIT PARAMETERS

The core parameters fed into the computer program in Appendix E are those of Core E-6.

The parameters for computing $\dot{\phi}_{ma}$ [Eqs. (18), (19), and (21)] at T = 29°C are as follows:

l_i = 22.19 mm; l_o = 23.54 mm;

ϕ_r = 3.45 maxwells; ϕ_s = 3.726 maxwells;

H_a = 310 amp-turns/m; H_q = 35.0 amp-turns/m;

H_n = 30.0 amp-turns/m; F_0'' = 0.95 amp-turn;

F_0 = 1.45 amp-turns; F_B = 3.12 amp-turns;

ν = 1.3; λ = 0.069 ohm/turn$^{2.3}$amp$^{0.3}$;

ρ_p = 0.113 ohm/turn^2.

The values of these parameters are as given in Report 3, p. 23, except for a correction in the value of H_a and very minor corrections in the values of ϕ_s, ν, ρ_p, and F_B. The corrections have been introduced as a result of more careful measurements of core parameters.

For computation of $\dot{\phi}_\epsilon$ [Eq. (2)], substitution of the values of l_i, l_o, ϕ_r, ϕ_s, and H_a into Eq. (3) gives ϵ = 0.3895 mμhy/turn^2.

Among the core parameters for computing $\dot{\phi}_i$ [Eq. (5)], it was found that F_{0i} = 0.55 amp-turn and ν_i = 1.3. The values of λ_i and, especially, C_i, were found to be dependent on the rise time of $F(t)$. For T_r in the neighborhood of 0.1 μsec, λ_i = 0.012 ohm/turn$^{2.3}$amp$^{0.3}$ and C_i = 0.245 amp-turn-μsec; for T_r around 0.02 μsec, λ_i = 0.014 ohm/turn$^{2.3}$amp$^{0.3}$ and C_i = 0.145 amp-turn-μsec. The parameters F_{0i}, ν_i, λ_i and C_i were determined by a cut-and-try method, in which the difference between experimental $\dot{\phi}(t)$ and computed ($\dot{\phi}_\epsilon + \dot{\phi}_{ma}$) was in reasonable agreement with the assumed $\dot{\phi}_i$ model, Eq. (5), regardless of the mmf-amplitude value, F_D. Specifically, eight values of F_D, varying from 0.6 amp-turn to 2.4 amp-turns, were examined.

The circuit parameters in the computer program merely describe the waveform of the drive mmf, $F(t)$. These are fed into the program via input-data cards, and include values for I_D (since a single-turn drive winding was used, $F_D = N_D I_D = I_D$), t_m, and S_p [*cf.* Part D-2(b)].

3. Results

Experimental and computed $F(t)$ and $\dot{\phi}(t)$ waveforms during the beginning of switching are compared in Fig. 10 for the drive mmfs with T_r near 0.1 μsec. Eight values of amplitude F_D were applied: 0.6, 0.8, 0.9, 1.0, 1.2, 1.5, 2.0, and 2.4 amp-turns. For each F_D value, two sets of experimental and computed $F(t)$ and $\dot{\phi}(t)$ waveforms are compared, using different time scales: Set i (on the left side of Fig. 10) emphasizes $\dot{\phi}(t)$ during the rise time of $F(t)$ (using a time scale of 40 nsec/div); Set ii (on the right) emphasizes $\dot{\phi}(t)$ immediately following the rise of $F(t)$.

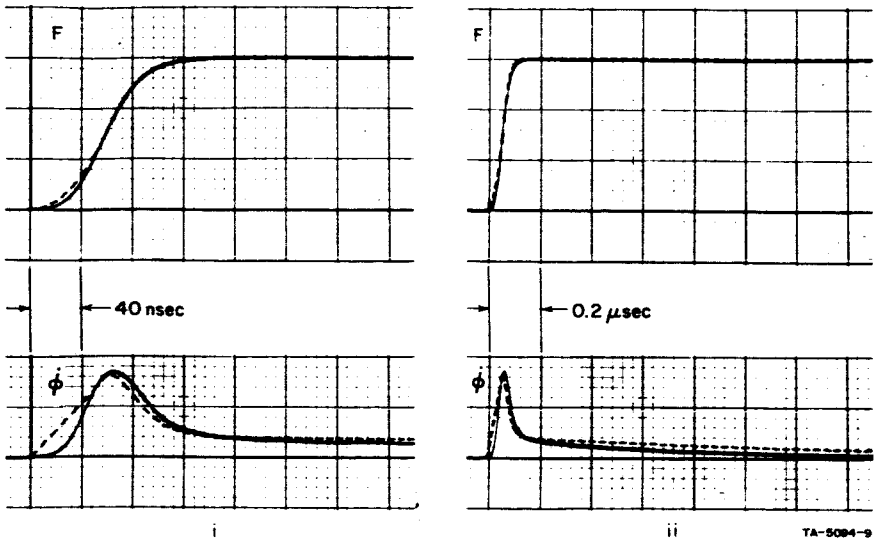
The solid lines in Fig. 10 are experimental oscillograms and the dashed lines are computed curves. These waveforms were produced without any manual drafting in the following manner: Negative enlargements of the original white on black experimental oscillograms were made on (transparent) acetate sheets, and the resulting scales of time, F, and $\dot{\phi}$ were inserted into the computer program (Appendix E). The program was run on a Burroughs B-5500 digital computer, and the results, first written on a magnetic tape, were plotted automatically as dashed lines by a CalComp Model 570 plotter. These computed plots and the enlarged negatives of the experimental oscillograms were then superimposed and photographed.

A comparison between experimental and computed $F(t)$ and $\dot{\phi}(t)$ waveforms for the drive mmfs with $T_r \cong 0.02$ μsec is shown in Fig. 11. Seven values of F_D were used: 0.6, 0.8, 0.9, 1.0, 1.18, 1.5, and 2.0 amp-turns. As in Fig. 10, for each F_D value, the emphasis is on $\dot{\phi}(t)$ during the rise of $F(t)$ in Set i, whereas Set ii shows the detail of $\dot{\phi}(t)$ following the rise of $F(t)$. As explained previously (*cf.* Fig. 8) the waveforms of $\dot{\phi}(t)$ are distorted slightly by ringing, especially during the rise of $F(t)$, shown in Set i. The photographic technique for producing Fig. 11 was the same as that used for Fig. 10.

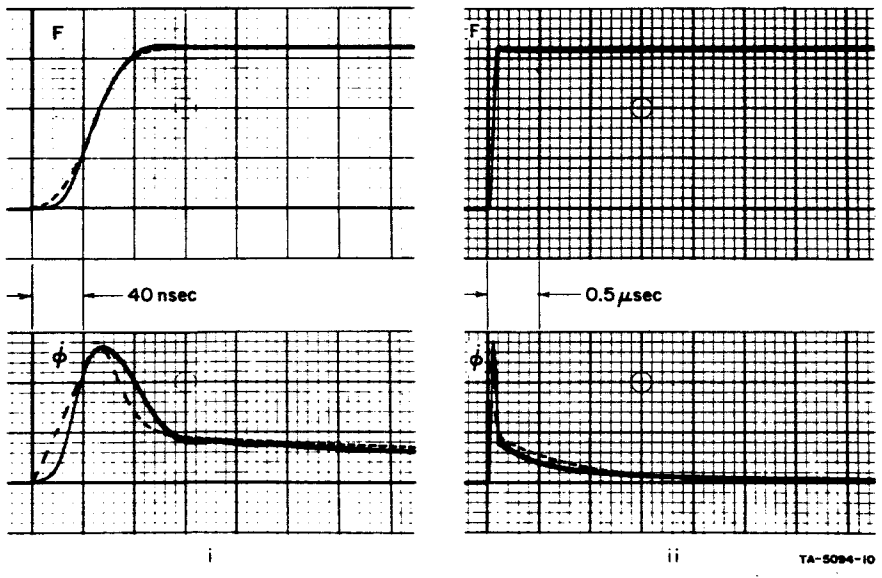
4. Discussion

a. Validity of the Models During the Beginning of Switching

The results in Figs. 10 and 11 show that, in general, there is a satisfactory agreement between experimental $\dot{\phi}(t)$ waveforms and the switching models for the three components of $\dot{\phi}$ proposed in this report. There are, however, some disagreements that need explanation and further investigation. Only a small portion of these disagreements stems from the differences between the actual $F(t)$ waveforms and the ones assumed in Eqs. (32).

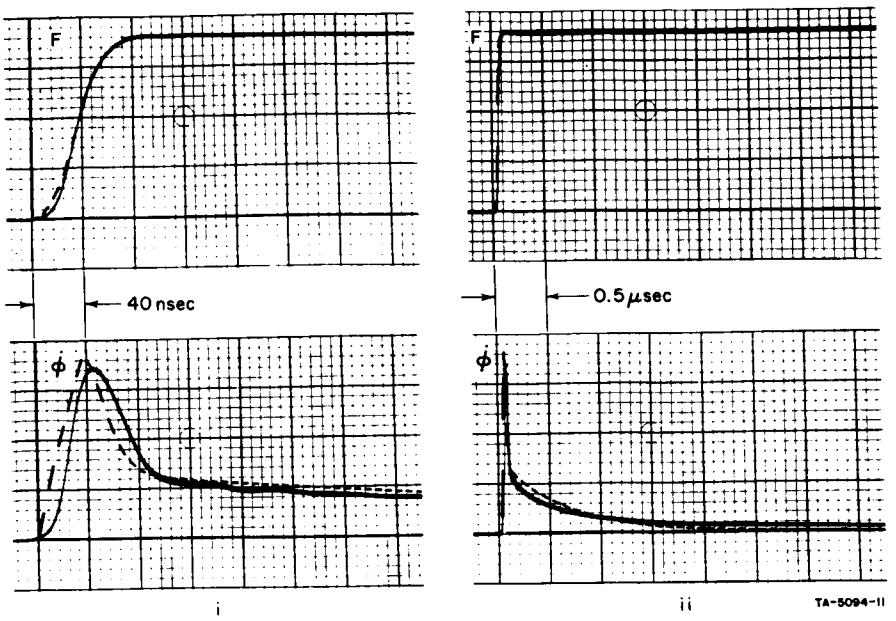


(a) F_D = 0.6 amp-turn
F scale = 0.2 amp-turn/major division
$\dot{\phi}$ scale = 2.5 millivolts/turn/major division

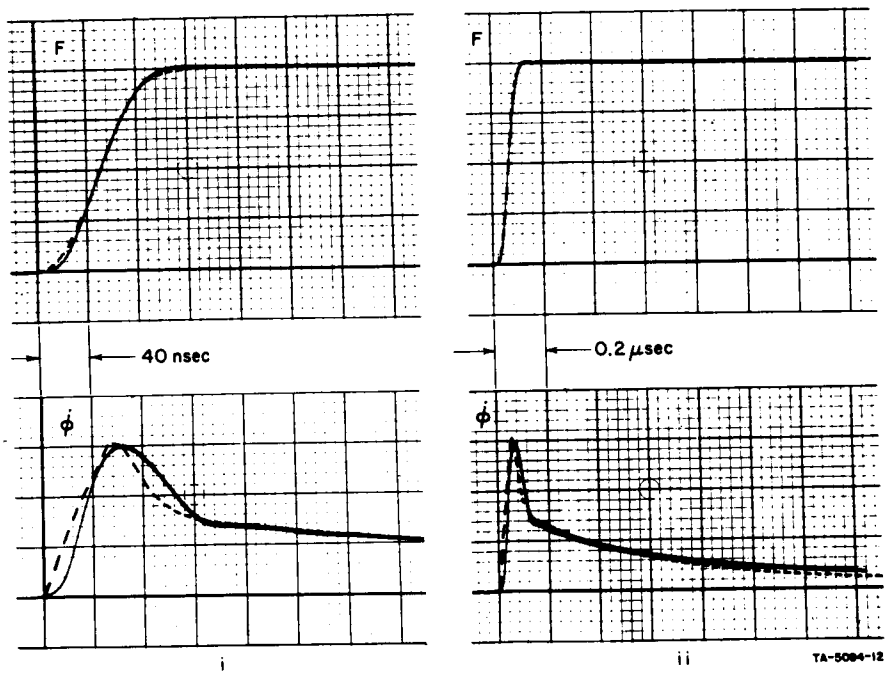


(b) F_D = 0.8 amp-turn
F scale = 0.25 amp-turn/major division
$\dot{\phi}$ scale = 2.5 millivolts/turn/major division

FIG. 10 EXPERIMENTAL (SOLID LINE) AND COMPUTED (DASHED LINE) F(t) AND $\dot{\phi}$(t) WAVEFORMS OF UNLOADED CORE E-6 DURING BEGINNING OF SWITCHING, USING F(t) WITH T_r OF ABOUT 0.1 μsec

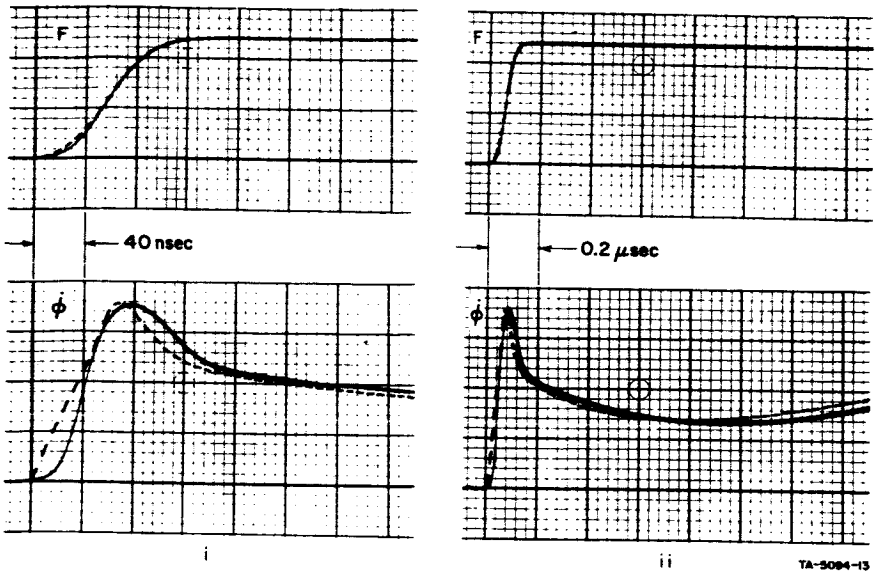


(c) F_D = 0.9 amp-turn
F scale = 0.25 amp-turn/major division
$\dot{\phi}$ scale = 2.5 millivolts/turn/major division

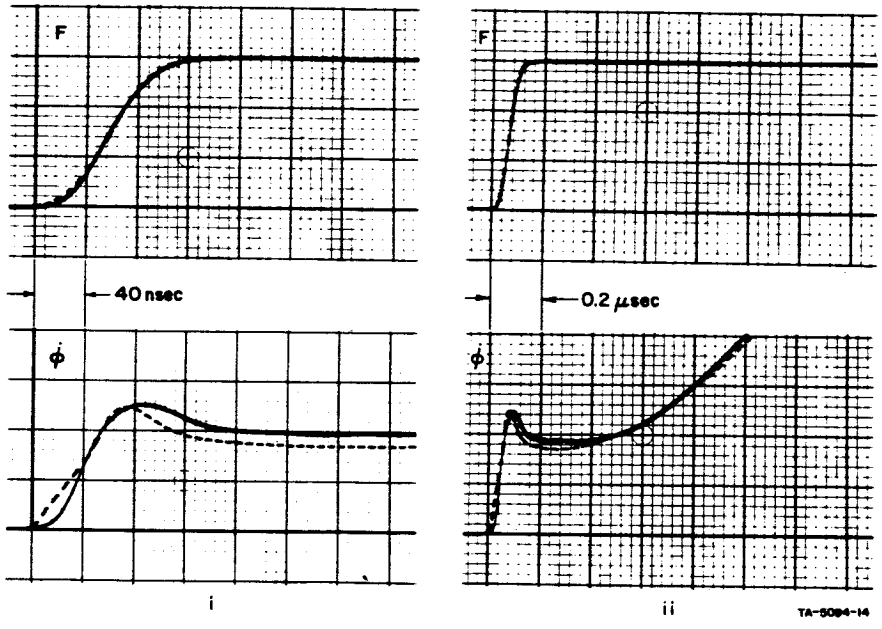


(d) F_D = 1.0 amp-turn
F scale = 0.25 amp-turn/major division
$\dot{\phi}$ scale = 2.5 millivolts/turn/major division

Fig. 10 Continued

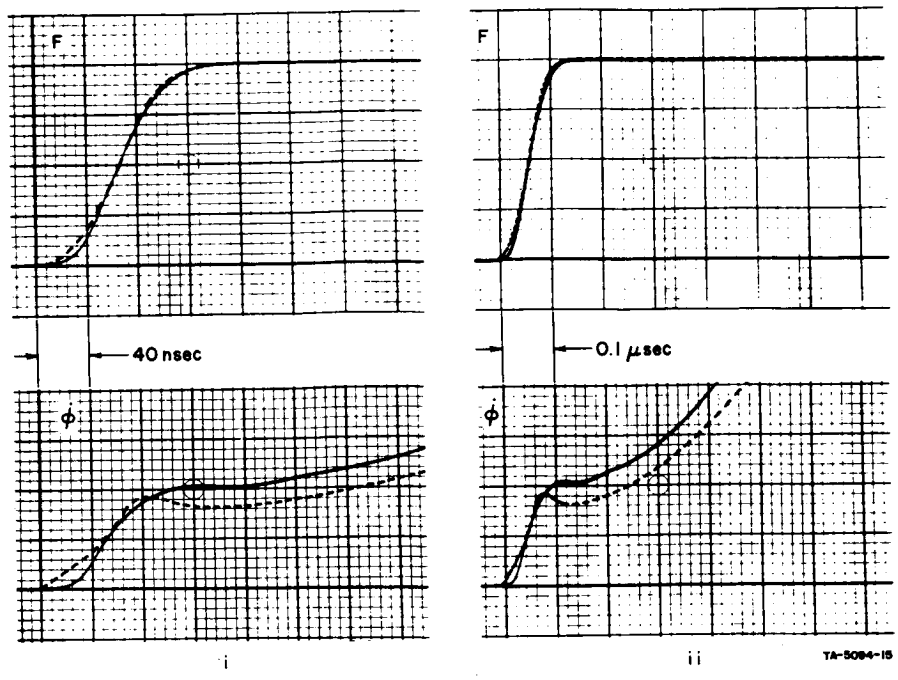


(e) F_D = 1.2 amp-turn
F scale = 0.5 amp-turn/major division
$\dot{\phi}$ scale = 2.5 millivolts/turn/major division

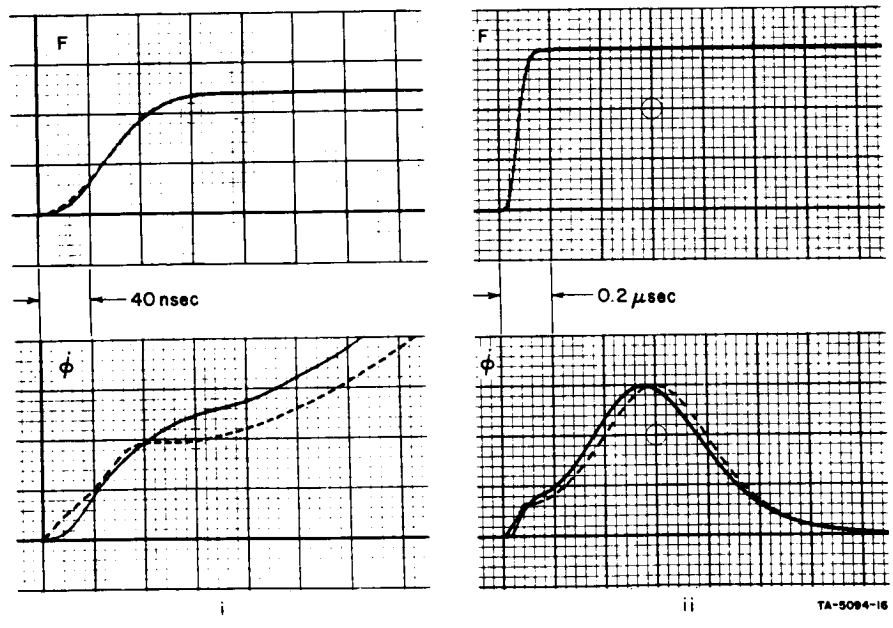


(f) F_D = 1.5 amp-turn
F scale = 0.5 amp-turn/major division
$\dot{\phi}$ scale = 5.0 millivolts/turn/major division

Fig. 10 Continued

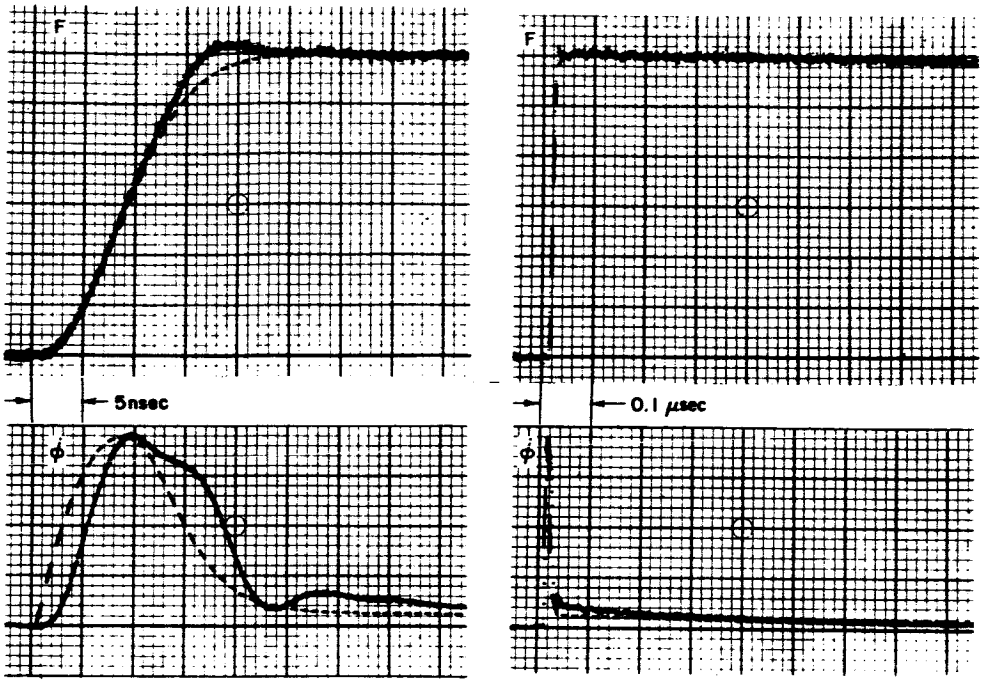


(g) F_D = 2.0 amp-turns
F scale = 0.5 amp-turn/major division
$\dot{\phi}$ scale = 10.0 millivolts/turn/major division

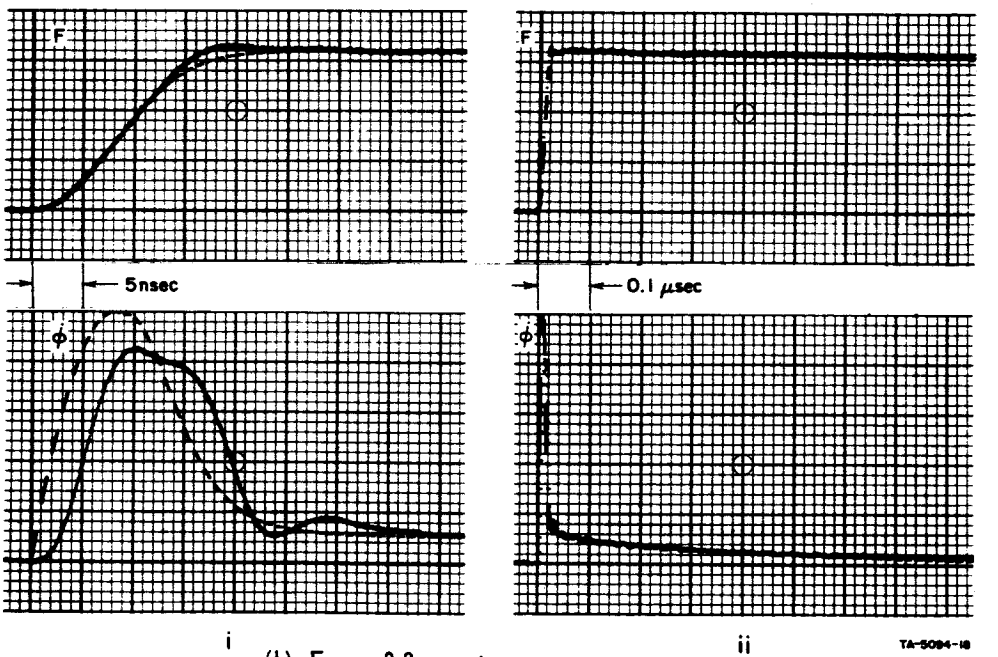


(h) F_D = 2.4 amp-turns
F scale = 1.0 (for i) and 0.75 (for ii) amp-turn/major division
$\dot{\phi}$ scale = 12.5 (for i) and 37.5 (for ii) millivolts/turn/major division

Fig. 10 Concluded

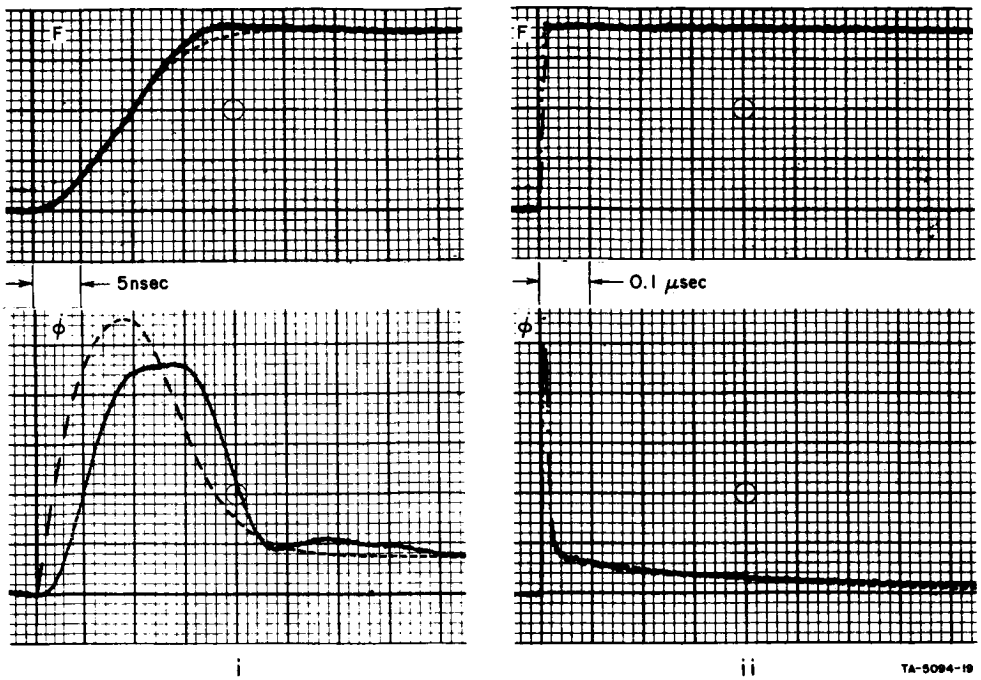


(a) F_D = 0.6 amp-turn
F scale = 0.1 amp-turn/major division
$\dot{\phi}$ scale = 5.0 millivolts/turn/major division

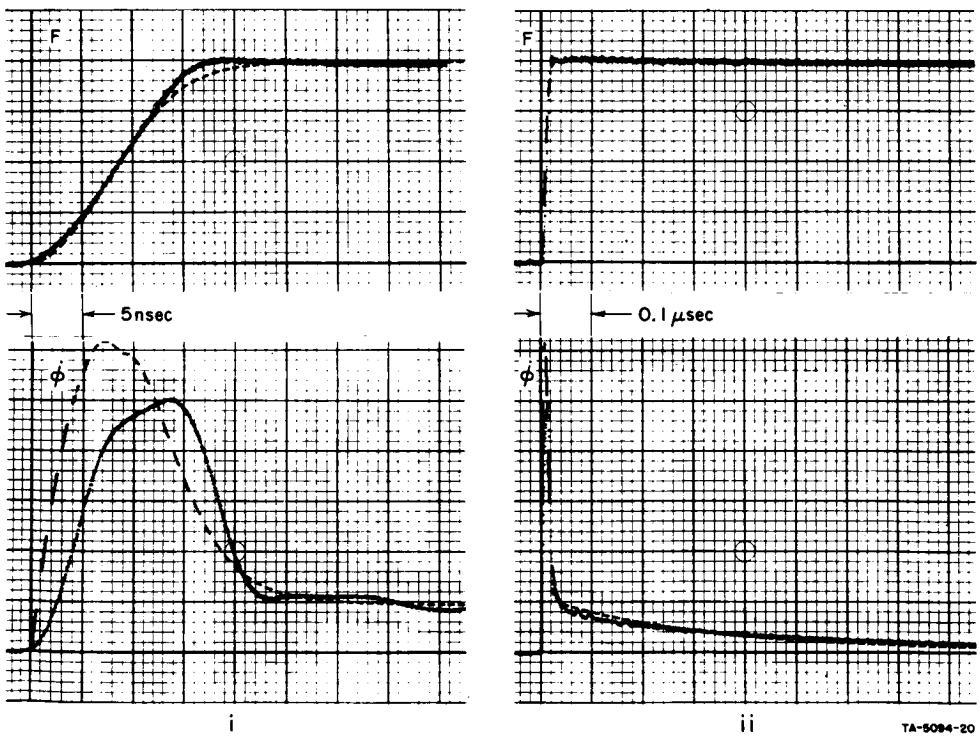


(b) F_D = 0.8 amp-turn
F scale = 0.25 amp-turn/major division
$\dot{\phi}$ scale = 5.0 millivolts/turn/major division

FIG. 11 EXPERIMENTAL (SOLID LINE) AND COMPUTED (DASHED LINE) F(t) AND $\dot{\phi}(t)$ WAVEFORMS OF UNLOADED CORE E-6 DURING BEGINNING OF SWITCHING, USING F(t) WITH T_r OF ABOUT 0.02 μsec

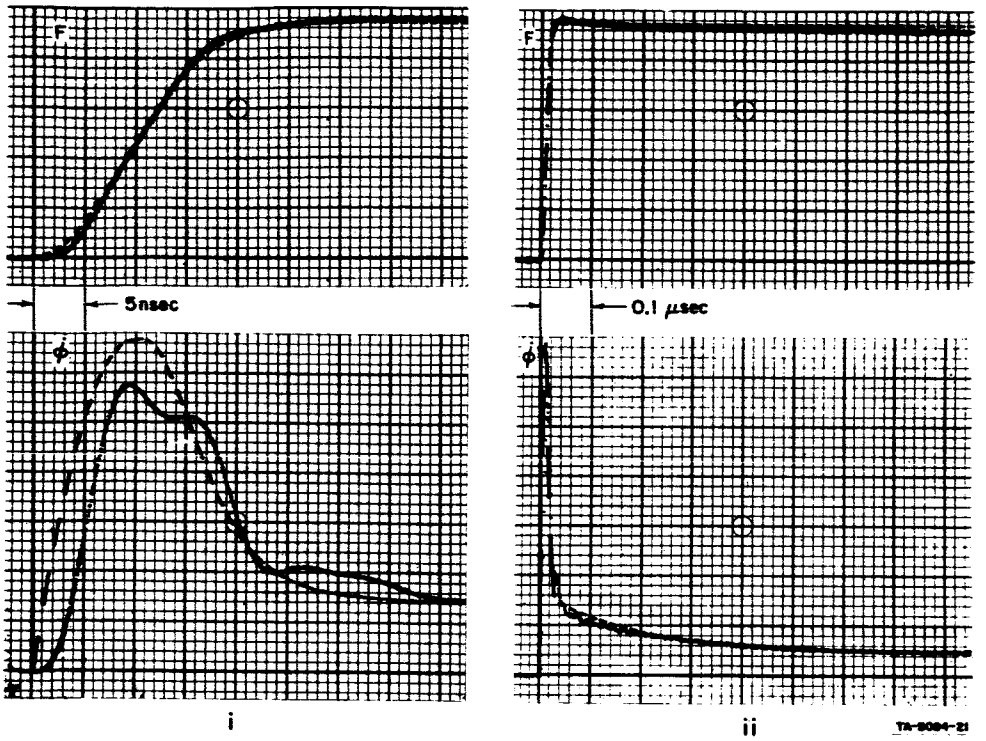


(c) F_D = 0.9 amp-turn
F scale = 0.25 amp-turn/major division
ϕ scale = 5.0 millivolts/turn/major division

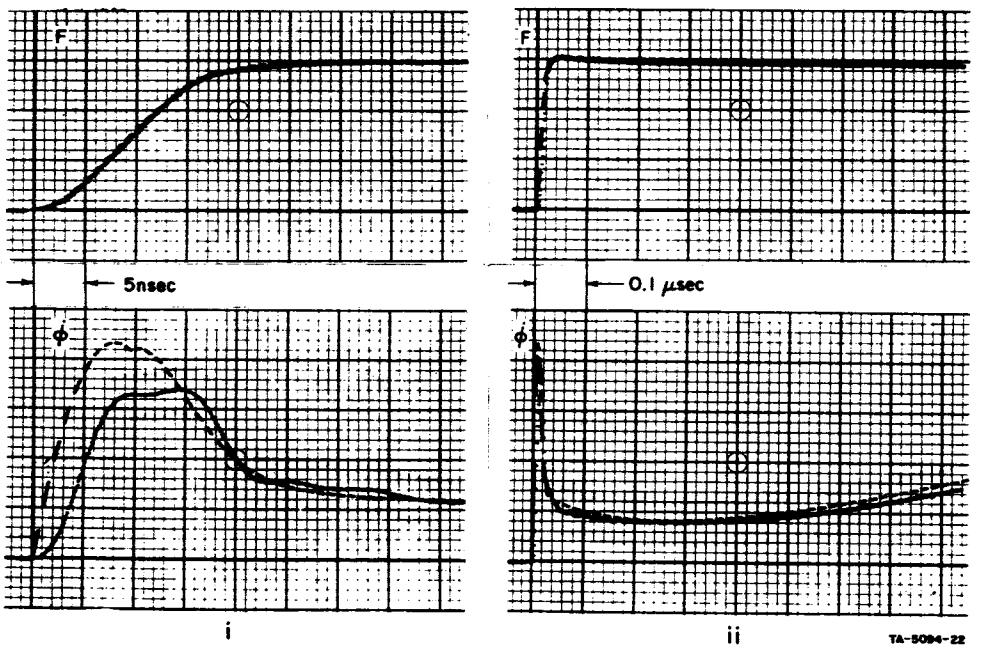


(d) F_D = 1.0 amp-turn
F scale = 0.25 amp-turn/major division
ϕ scale = 5.0 millivolts/turn/major division

Fig. 11 Continued

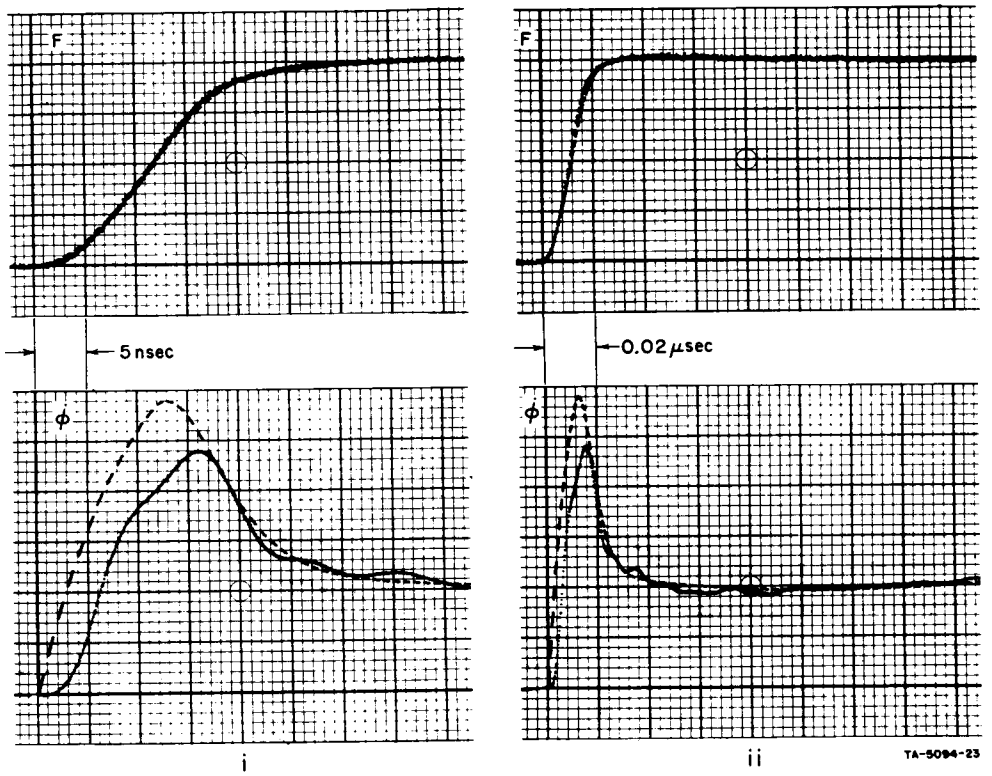


(e) F_D = 1.18 amp-turn
F scale = 0.25 amp-turn/major division
$\dot{\phi}$ scale = 5.0 millivolts/turn/major division



(f) F_D = 1.5 amp-turn
F scale = 0.5 amp-turn/major division
$\dot{\phi}$ scale = 10.0 millivolts/turn/major division

Fig. 11 Continued



(g) F_D = 2.0 amp-turns
F scale = 0.5 amp-turn/major division
$\dot{\phi}$ scale = 10.0 millivolts/turn/major division

Fig. 11 Concluded

The model for the elastic component of $\dot{\phi}$, $\dot{\phi}_\epsilon = \epsilon\dot{F}$, turns out to be too simple. It is based on the assumption that the elastic motion of domain walls and elastic rotation of magnetization encounter only pull-back (or stiffness) forces that are proportional to the relatively small displacement and small angle of rotation of magnetization. There is, however, a short delay between $\epsilon\dot{F}$ and the actual $\dot{\phi}_\epsilon$. This delay is probably caused by viscous damping that is proportional to $\dot{\phi}_\epsilon$. Inclusion of the viscous damping will result in a differential equation of the form

$$\delta\ddot{\phi}_\epsilon + \dot{\phi}_\epsilon = \epsilon\dot{F} \quad , \tag{35}$$

where δ is a constant proportional to the viscous damping.

The model for $\dot{\phi}_i$, Eq. (5), appears to agree quite well with experimental $\dot{\phi}(t)$ during the beginning of switching. It should be emphasized that only one core (Core E-6) has so far been investigated thoroughly. The $\dot{\phi}(t)$ waveforms of other tested cores were similar to those of Core E-6; however, these waveforms have not yet been analyzed quantitatively. Clearly, such an investigation, although costly, should be performed with other materials in order to substantiate the proposed $\dot{\phi}_i$ model.

b. VARIATIONS IN SWITCHING PARAMETERS

We have seen that in order to obtain a satisfactory agreement between observed $\dot{\phi}(t)$ waveforms and the model for $\dot{\phi}_i$, the switching parameters λ_i and C_i had to depend on the rise time, T_r, of $F(t)$. A slightly larger λ_i and a smaller C_i correspond to a shorter T_r. This dependence of λ_i and C_i on T_r is very complex, and may, perhaps, be explained qualitatively as follows. If T_r is short, domains in *some* regions would expand only *locally* when $F(t) = F_D$, and thus contribute a $\dot{\phi}_i$ component of a short duration. If T_r is long, such local domain expansions are given no chance to occur: before F reaches the local threshold values, flux reversal in these regions is caused by the applied F plus the magnetic poles brought about by oncoming unobstructed domain walls. The shorter T_r is, the larger is the number of nucleation centers around which short-lived, local, domain-wall motions take place, and thus the larger is the mean displacement time of these walls. This argument may explain why λ_i increases and C_i decreases as T_r decreases.

It is interesting to note that $\nu_i = \nu$. Whether this is a coincidence or not remains to be determined after additional core materials are investigated. Some light may be shed on this problem by attempting to explain the physical origin of ν. Conger and Essig hypothesized[6] that the threshold field for domain-wall motion in thin films is distributed randomly over the range between the coercive force and the anisotropy field. As a result, the number of walls, in addition to the wall velocity, increases with H, and the nonlinearity of $1/\tau_s$ *vs.* H is accounted for. In a similar way, by assuming that $\dot{B}_p = \kappa(H - H_0'')^\nu$ [Report 2, p. 38, Eq. (83)], an effective H_0, denoted by H_0', that increases with H between H_0'' and H_B [Report 2, p. 38, Fig. 15] was obtained. The motion of locally obstructed domain walls is governed by the same rule. If the value of ν is determined by the distribution function of the threshold field, and if the same distribution function is applicable to F_i, one would expect ν_i and ν to be equal.

We have seen that there is an analogy between the parameters λ, F_0'', and ν of $\dot{\phi}_{ma}$ and the parameters λ_i, F_i, and ν_i of $\dot{\phi}_i$. Let us carry this analogy a little further. Referring to Eq. (19), $\dot{\phi}_p = \rho_p(F - F_0)$, where $F_0 > F_0''$, if $F_B \leq F$. Hence, we expect that for high F values, compared with those in Figs. 10 and 11, $\dot{\phi}_i$ will become proportional to $(F - F_d^n)$, where $F_d^n > F_{0i}$. This is in agreement with Eq. (4) and with the fact that the ratio $F_0/F_0'' = 1.45/0.95 = 1.52$ is close to the ratio $F_d^n/F_{0i} = 0.775/0.55 = 1.41$ (*cf.* Report 3, p. 29). In order to verify this conclusion, a drive current with higher F_D values and shorter rise time than in Fig. 11 should be used in the experiment (if T_r is not decreased, $\dot{\phi}_{ma}$ will mask $\dot{\phi}_i$). With our present laboratory equipment, this could be achieved by using a mercury-contact switch and a transmission line. As explained previously, when such an attempt was made, the signal-to-noise ratio was too low to be analyzed meaningfully. We hope to overcome this circuit problem in the future.

c. VARIATION OF $\dot{\phi}(t)$ WITH F_D

In Fig. 10, individual $\dot{\phi}(t)$ waveforms are shown for the different values of F_D. The effect of the magnitude of F_D on the waveform of $\dot{\phi}(t)$ is demonstrated qualitatively in Fig. 12 by showing a multiple exposure of experimental $\dot{\phi}(t)$ oscillograms. The only difference between Fig. 12(a) and Fig. 12(b) is the time scale and the duration of switching. In either figure, the $\dot{\phi}(t)$ oscillograms correspond to $F(t)$ drives of the same rise

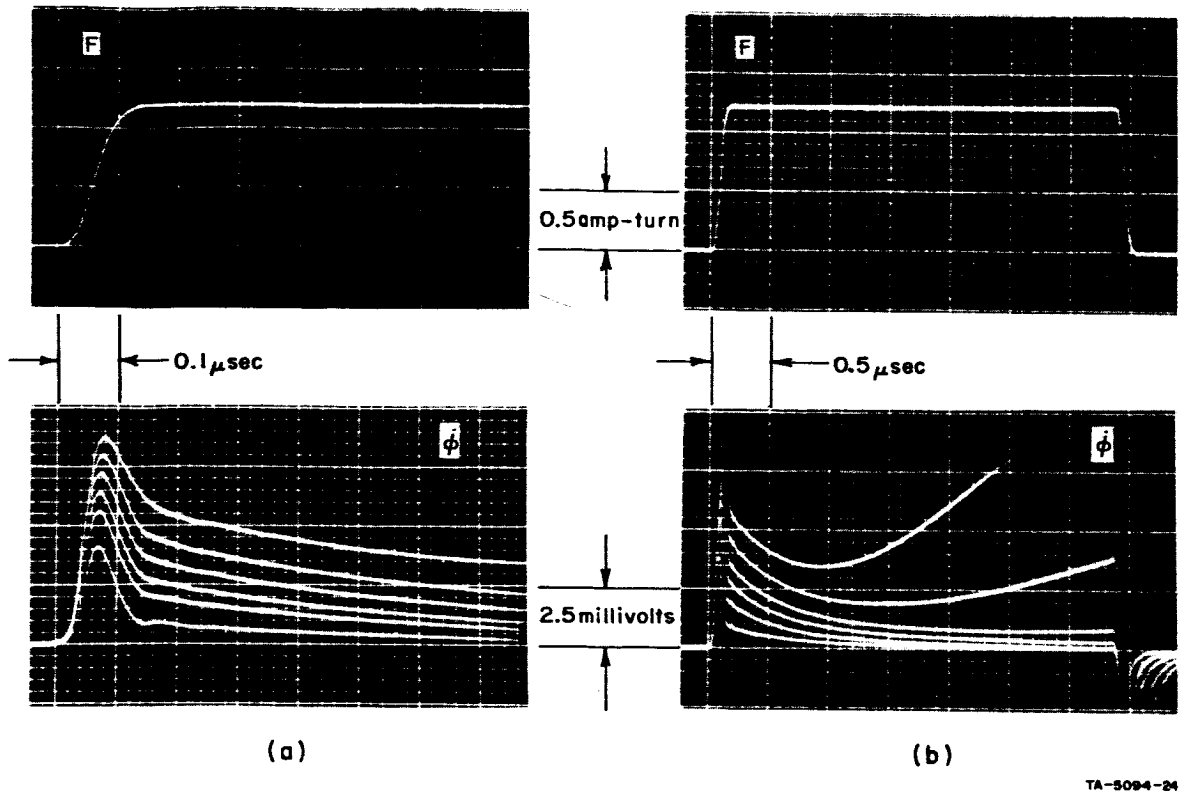


FIG. 12 MULTIPLE EXPOSURE OF $\dot{\phi}(t)$ WAVEFORMS OF CORE E-6 CORRESPONDING TO F(t) PULSES OF DIFFERENT AMPLITUDES

time and the following F_D values: 0.6, 0.8, 0.9, 1.0, 1.1, and 1.2 amp-turn; the $F(t)$ oscillogram shown above the $\dot{\phi}(t)$ oscillograms corresponds to F_D = 1.2 amp-turn.

d. COMPUTED COMPONENTS OF $\dot{\phi}(t)$

The $\dot{\phi}(t)$ waveforms shown in Figs. 10 and 11 are those of the *total* $\dot{\phi}$. The relative magnitudes of the three components of $\dot{\phi}(t)$ depend on $F(t)$. For example, computed $\dot{\phi}(t)$ and its components are shown in Fig. 13 for the case of F_D = 2.0 amp-turn and T_r = 0.024 μsec. This information about $\dot{\phi}_\epsilon(t)$, $\dot{\phi}_i(t)$, and $\dot{\phi}_{ra}(t)$ is a part of the computer output (*cf.* Appendix E).

e. COMPUTED $\dot{\phi}_{ip}$ AND T_{ip} *vs.* F_D AND T_r

In Sec. IB-2(d), we have calculated $\dot{\phi}_i(t)$ [Eqs. (8)], and its peak amplitude $\dot{\phi}_{ip}$ [Eq. (11)] for the case of $F(t)$ with a ramp rise

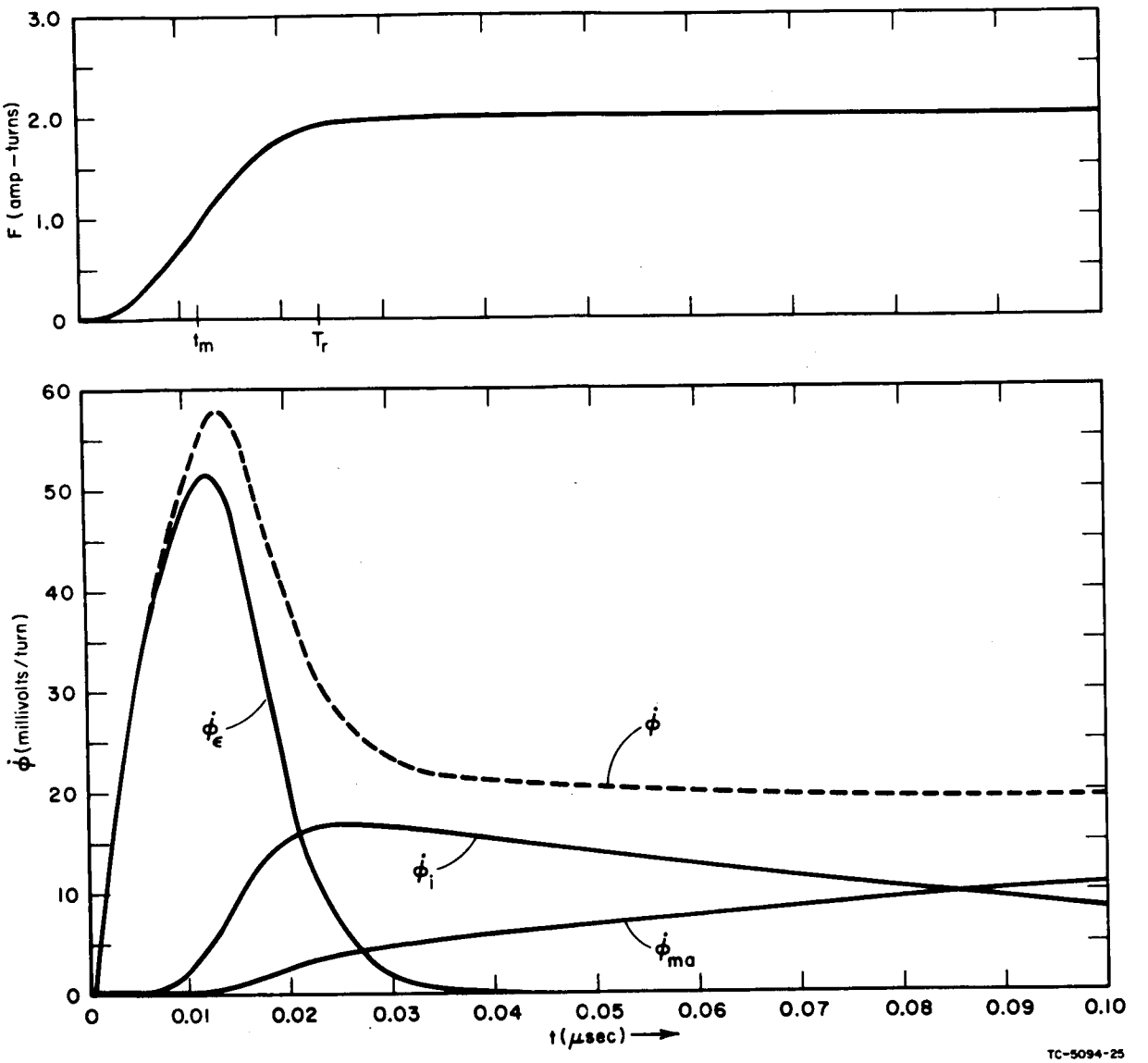


FIG. 13 COMPUTED $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$ AND $\dot{\phi}$ vs. t OF CORE E-6 DURING THE BEGINNING OF SWITCHING

followed by a constant amplitude, Eq. (6). Similar computation was performed for the drive current expressed by Eqs. (32) as part of the computed $\dot{\phi}(t)$ in Figs. 10 and 11. Plots of $\dot{\phi}_{ip}$ *vs.* F_D are shown in Fig. 14 for three values of rise time: $T_r \cong 0.1$ μsec, $T_r \cong 0.02$ μsec, and $T_r = 0$. The small variations around a smooth curve in each of the first two cases stem from the small variations in the actual values of T_r. For the case of $T_r = 0$, $T_{ip} = 0$ and $\dot{\phi}_{ip} = \lambda_i (F_D - F_i)^{\nu_i}$ [Eq. (5)], where $F_i = F_{0i} \tanh (F_D/F_{0i})$ [Eq. (17)] and $\lambda_i = 0.014$ ohm/turn$^{2.3}$amp$^{0.3}$.

e. ADDITIONAL LOW $\dot{\phi}_i$ COMPONENT

If $F \leq F_0''$, then $\dot{\phi}_{ma} = 0$ and so

$$\dot{\phi} = \dot{\phi}_\epsilon + \dot{\phi}_i \quad . \tag{36}$$

Suppose that F is a rectangular pulse of amplitude F_D and duration T. Let us examine the amount of $\Delta\phi$ contributed by $\dot{\phi}_\epsilon$ and $\dot{\phi}_i$ by the time $t = T$, before F begins to fall. Since $\dot{\phi}_\epsilon = \epsilon \dot{F}$,

$$\Delta\phi_\epsilon = \int_0^{F_D} \epsilon dF = \epsilon F_D \quad . \tag{37}$$

Following Eq. (5),

$$\Delta\phi_i = \int_0^T \dot{\phi}_i dt = \lambda_i C_i (F_D - F_i)^{\nu_i - 1} [1 - e^{-T(\dot{F} - F_i)/C_i}] \quad . \tag{38}$$

As $T \to \infty$, $\Delta\phi_i \to \Delta\phi_{i(\infty)}$, where

$$\Delta\phi_{i(\infty)} = \lambda_i C_i (F_D - F_i)^{\nu_i - 1} \quad . \tag{39}$$

Adding $\Delta\phi_\epsilon$ and $\Delta\phi_{i(\infty)}$, the total $\Delta\phi$ due to a step F, $F = F_D$, is

$$\Delta\phi_{(\infty)} = \epsilon F_D + \lambda_i C_i (F_D - F_i)^{\nu_i - 1} \quad . \tag{40}$$

Let us calculate $\Delta\phi_{(\infty)}$ of Core E-6, assuming an F_D value which is less than $F_0'' = 0.95$ amp-turn, *e.g.*, $F_D = 0.9$ amp-turn. The core parameters are as follows: $\epsilon = 0.3895$ mμhy/turn^2, $\nu_i = 1.3$,

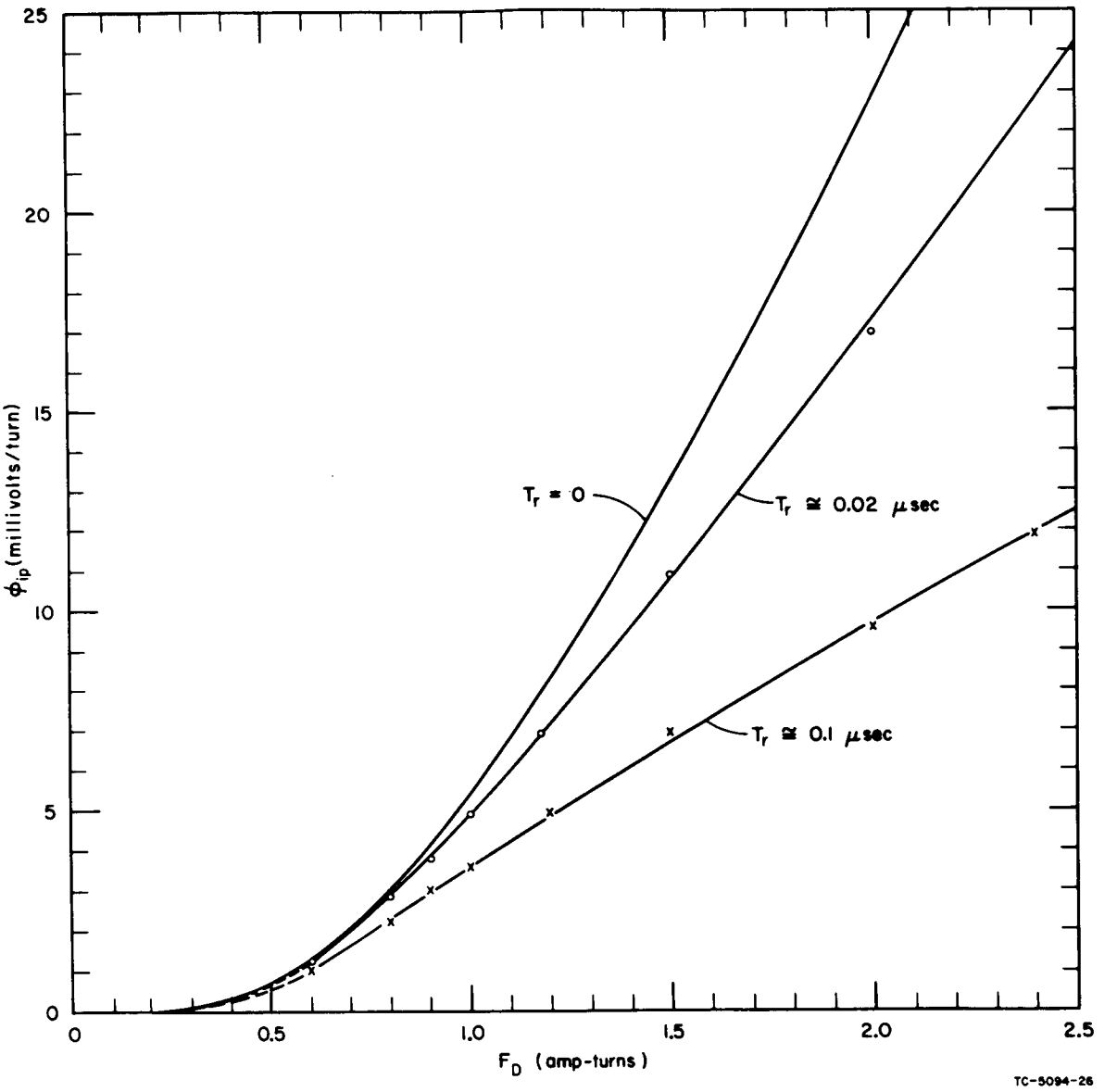


FIG. 14 COMPUTED $\dot{\phi}_{ip}$ vs. F_D OF CORE E-6 WITH T_r AS A PARAMETER

λ_i = 0.014 ohm/turn$^{2.3}$amp$^{0.3}$, C_i = 0.145 amp-turn-μsec (assumed to be same as for $T_r \cong 0.02$ μsec), and $F_i \cong 0.55$ amp-turn. Substituting these values into Eqs. (37), (39) and (40), we find that $\Delta\phi_\epsilon$ = 0.034 maxwell, $\Delta\phi_{i(\infty)}$ = 0.148 maxwell and $\Delta\phi_{(\infty)}$ = 0.182 maxwell.

Now consider the static $\phi(F)$ curve. Starting from $\phi = -\phi_r$, the total flux change involved in reaching a point (F, ϕ_d) on this curve is

$$\Delta\phi_d = \phi_d + \phi_r \quad . \tag{41}$$

For Core E-6, $F = F_D$ = 0.9 amp-turn corresponds to ϕ_d = −0.12 maxwell, and since ϕ_r = 3.45 maxwells, $\Delta\phi_d$ = 3.33 maxwells.

Comparing $\Delta\phi_d$ and $\Delta\phi_{(\infty)}$, we find that (for Core E-6) $\Delta\phi_d/\Delta\phi_{(\infty)} = 3.3/0.182 = 18.3$. We conclude from this result that the model for ϕ_i given in Eq. (5) does not account for the very slow flux switching involved in reaching a point on the static $\phi(F)$ curve. This conclusion is in agreement with the $\dot{\phi}(t)$ waveforms in Sets (a-ii), (b-ii) and (c-ii) of Figs. 10 and 11. One can detect that after the computed $\dot{\phi}$ has decayed to a negligible value, a very low experimental $\dot{\phi}$ continues to exist. Physically, such a low $\dot{\phi}$ results from the motion of *few* domain walls that are obstructed far from their original position rather than locally. If this interpretation is valid, this very low $\dot{\phi}$ may be considered as an additional component of $\dot{\phi}_i$ whose amplitude is much smaller than $\lambda_i(F - F_i)^{\nu_i}$ and whose time constant is so much larger than $C_i/(F - F_i)$ that the product of its amplitude and its time constant is much larger than $\lambda_i C_i (F - F_i)^{\nu_i - 1}$. An alternative source of this very low $\dot{\phi}$ to be considered is the main $\dot{\phi}$ component, $\dot{\phi}_{ma}$, with F_0'' decreasing for decreasing F [similarly to F_i, Eq. (17)]. Future investigation of this very low $\dot{\phi}_i$ component should be based on correlation between measured $\phi(F)$ curves obtained by rectangular F pulses of different duration T and calculated $\Delta\phi_d - \Delta\phi_{(\infty)}$.

5. Summary

Verification of the models for the three components of $\dot{\phi}(t)$ was made by comparing experimental $\dot{\phi}(t)$ and computed $\dot{\phi}(t) = \dot{\phi}_\epsilon(t) + \dot{\phi}_i(t) + \dot{\phi}_{ma}(t)$ during the beginning of the flux switching. The tested core was a thin magnesium-manganese-zinc ferrite ring of OD/ID = 1.06 (Core E-6, Report 3, p. 23; F_c = 0.9 amp-turn). The comparison was made for mmf pulses of different amplitude, F_D, and rise time, T_r: eight values of F_D (ranging from 0.6 to 2.4 amp-turns) for $T_r \cong 0.1$ μsec and seven values of F_D

(ranging from 0.6 to 2.0 amp-turns) for $T_r \cong 0.02$ μsec. Each case was repeated twice, using different time scales. The core and circuit parameters were fed into a computer program which computed $\dot{\phi}_\epsilon(t)$, $\dot{\phi}_i(t)$, $\dot{\phi}_{ma}(t)$ and $\dot{\phi}(t)$. The resulting $\dot{\phi}(t)$ waveforms of the above 2 × 15 cases were machine plotted, and are compared with the experimental oscillograms in Figs. 10 and 11. From this comparison, the following observations are made relative to Core E-6:

(1) There is a small delay (of the order of 0.15 T_r) between computed $\epsilon\dot{F}(t)$ and experimental $\dot{\phi}_\epsilon(t)$ due, probably, to viscous damping. An improved model for elastic switching is $\delta\ddot{\phi}_\epsilon + \dot{\phi}_\epsilon = \epsilon\dot{F}$, where δ is a constant proportional to the viscous damping.

(2) The parameters λ_i and C_i depend on T_r: as T_r decreases, λ_i increases (slightly) and C_i decreases.

(3) The analogy between $\dot{\phi}_p = \lambda(F - F_0)^\nu$ and $\dot{\phi}_{ip} = \lambda_i(F - F_i)^{\nu_i}$ agrees with the analogy between $\dot{\phi}_p = \rho_p(F - F_0)$ and $\dot{\phi}_{ip} = \rho_i(F - F_d^n)$ that was proposed in Report 3. Furthermore, $\nu = \nu_i$ and the ratio F_0/F_0'' is close to the ratio F_d^n/F_{0i}. It is expected, therefore, that Eq. (4) is valid for high values of F_D. This may be verified by applying current pulses of high amplitude and very short rise time, such as obtained by using a mercury-relay pulser.

(4) The flux change $\Delta\phi_{i(\infty)} = \int_0^\infty \dot{\phi}_i dt$ is much smaller than the flux change, $\Delta\phi_d = \phi_d + \phi_r$, involved in obtaining a point on the static $\phi(F)$ curve. In order to account for the latter, another term should be added to the expression for $\dot{\phi}_i$, Eq. (5). This term is expected to be of the same form as the existing expression, except for a much lower amplitude and a very much longer time constant. Alternatively, most of $\Delta\phi_d$ may be accounted for by $\dot{\phi}_{ma}$ if F_0'' becomes lower for low F values.

E. CONCLUSIONS

For the type of ferrite material investigated so far, the $\dot{\phi}(t)$ waveform resulting from a constant-amplitude mmf of a finite rise time consists of three components: elastic, decaying inelastic, and main (bell-shaped) inelastic.

The elastic $\dot{\phi}$ component is described by the model $\dot{\phi}_\epsilon = \epsilon \dot{F}$. This model results in a small delay between experimental and computed $\dot{\phi}(t)$. This delay implies that an improved model for $\dot{\phi}_\epsilon$ is of the form $\delta \ddot{\phi}_\epsilon + \dot{\phi}_\epsilon = \epsilon \dot{F}$, where δ is a constant proportional to the viscous damping.

The decaying $\dot{\phi}$ component, $\dot{\phi}_i$, is described by the model $\dot{\phi}_i = \lambda_i (F - F_i)^{\nu_i} \exp\left[-(t - T_i)(F - F_i)/C_i\right]$, where λ_i, F_{0i}, ν_i, and C_i are switching parameters and T_i is the time F reaches F_i.

The amount of $\Delta\phi$ associated with reaching a point on the static $\phi(F)$ curve for $F < F_0''$ is much larger than $\int_0^\infty \dot{\phi}_i dt$. In order to account for this $\Delta\phi$, the existing $\dot{\phi}$ model needs to be modified either by adding a component of the same form as $\dot{\phi}_i$, except of a much lower amplitude and of a much longer time constant, or by lowering F_0'' for low values of F, as was done for F_i.

The main component, $\dot{\phi}_{ma}$, may be expressed by the model $\dot{\phi}_{ma} = \dot{\phi}_p \eta(\phi)$, where $\dot{\phi}_p$ is the peak $\dot{\phi}$, and $\eta(\phi) = 1 - [(2\phi + \phi_r - \phi_d)/(\phi_r + \phi_d)]^2$, and where, for a given F value, ϕ_d is the ϕ value on the static $\phi(F)$ curve.

If F is large (compared with the coercive mmf, F_c), computation of $\dot{\phi}_i + \dot{\phi}_{ma}$ may be approximated by the $\dot{\phi}_{ma}$ model alone if ϕ_s replaces ϕ_r in the expression for $\eta(\phi)$. The elastic component, $\dot{\phi}_\epsilon$, may be neglected if only inelastic $\Delta\phi$ is of concern.

Further investigation is needed in the following areas:

(1) Improving the model for $\dot{\phi}_\epsilon$ by including the effect of viscous damping.

(2) Studying the effect of ϕ and switching history on ϵ.

(3) Extending the investigation of $\dot{\phi}_i$ to high values of F.

(4) Investigating the very slow $\dot{\phi}$ component associated with the static $\phi(F)$ curve.

(5) Repeating the investigation of the $\dot{\phi}$ components on other types of magnetic materials.

II COMPUTATION OF FLUX SWITCHING IN MAGNETIC CIRCUITS

In Report 3, the switching model for the main component of $\dot{\phi}$ was applied (using a computer program) and verified experimentally for three magnetic circuits: an unloaded core switched by step, ramp, triangular and trapezoidal mmfs; a loaded core driven by step and ramp drives; and a core-diode shift register. In this report, computer programs are provided and the results are compared with experimental data for the following four magnetic circuits:

(1) Unloaded core driven by $F(t)$ of constant-amplitude, F_D, and rise-time, T_r—computation *vs.* time of $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$, $\dot{\phi}$, and ϕ. (See Sec. I-D-2-c.)

(2) A core loaded by an inductive load (R-L, R-L-C, and R-L-C-diode) and driven by step and ramp mmf—computation *vs.* time of F, $\dot{\phi}$, ϕ, and the load current, using a more exact algorithm than in Report 3.

(3) A core-diode-transistor binary counter—computation (*vs.* time) of three currents and of F, $\dot{\phi}$ and ϕ of each of two cores.

(4) A loaded, saturable, three-leg core—computation *vs.* time of F, $\dot{\phi}$, and ϕ of each leg, and computation of division of flux between two legs in parallel as a function of drive current amplitude, load, and relative leg dimensions.

Item (1) is treated in Sec. I-D; Items (2), (3), and (4) are treated in the following sections.

A. LOADED CORE

1. Introduction

In Report 3, flux switching in a loaded core was analyzed (Report 3, pp. 39-55) and a computer program was provided (Report 3, pp. 149-153) for computation of $\dot{\phi}$, ϕ, and i_L (the load current) *vs.* time. The load consisted of six combinations of a resistance, an inductance, a capacitance, and a diode: R, R-diode, R-L, R-C, R-L-C, and R-L-C-diode. Computed and measured waveforms of $\dot{\phi}(t)$ and $i_L(t)$ were compared for

step and ramp drives. The algorithm used to compute these variables was the same regardless of the type of load. This resulted in accumulation of error in the cases of inductive load (R-L, R-L-C, and R-L-C-diode). Our main objective now is to improve the accuracy of computation for these cases by using a more exact algorithm. An additional objective is to explain why an iterative method of solving a transcendental equation (such as Newton's method) must be applied, as was done in Report 3, if $L = 0$.

Let us summarize briefly the analysis in Report 3 (Fig. 14, p. 39, and pp. 41-45). A drive current i_D is applied in N_D turns to a core which is coupled by N_c turns to a load. The load is composed of a resistance R_L, an inductance L, a capacitance C, and a diode. The diode voltage is expressed as

$$V_d = i_L R_d + E_k \ln\left(1 + \frac{i_L}{I_0}\right) , \tag{42}$$

where R_d is a forward resistance, E_k is a voltage constant and I_0 is saturation current. Since $F = N_D i_D - N_c i_L$ and i_D is a given function of t, $\dot{\phi}(F,\phi)$ may be written as $\dot{\phi}(t, i_L, \phi)$. Letting $R_L + R_d = R$ and $\int_0^t i_L dt = q$, the load loop equation is

$$\frac{1}{C} q + R i_L + L \frac{di_L}{dt} + E_k \ln\left(1 + \frac{i_L}{I_0}\right) - N_c \dot{\phi}(t, \phi, i_L) = 0 . \tag{43}$$

In Report 3, i_L was solved for transcendentally, using Newton's method of successive iterations. In each jth iteration, corrected values of $\dot{\phi}$ and $i_L = \dot{q}$ were used to correct the values of ϕ and q of the previous iteration, using Eq. (31) and its equivalence for q. Although this method worked well for all load cases, some error was accumulated at each nth Δt as a result of approximating $(di_L/dt)_n$ by $[i_{Ln} - i_{L(n-1)}]/\Delta t$ and $(d^2 i_L/dt^2)_n$ by $[(di_L/dt)_n - (di_L/dt)_{n-1}]/\Delta t$ [*cf.* Report 3, Eqs. (67), (69), and (70), pp. 44-45]. If $L \neq 0$, this error may be reduced by using a different method of evaluating i_{Ln} in each iteration. We thus distinguish between two cases: $L \neq 0$ and $L = 0$.

2. Inductive Load

If $L \neq 0$, there is no need for a transcendental solution of i_L. The switching function $\dot{\phi}(t, \phi, i_L)$ together with the second-order differential equation, Eq. (43), may be written as a set of three simultaneous differential equations of the first order:

$$\frac{d\phi}{dt} = \dot{\phi}(t, \phi, i_L) \qquad [44(a)]$$

$$\frac{dq}{dt} = i_L \qquad [44(b)]$$

$$\frac{di_L}{dt} = \frac{1}{L}\left[N_c\dot{\phi}(t,\phi, i_L) - \frac{1}{C}q - Ri_L - E_k ln\left(1 + \frac{i_L}{I_0}\right)\right] \quad . \qquad [44(c)]$$

In these equations, t is an independent variable, and ϕ, q, and i_L are dependent variables. Note that, as required by the various methods of numerical solutions of differential equations,[7] differentials appear on the left side of the equality signs only. The numerical solution of Eqs. (44) may be achieved by various well-known methods, such as Adam's, Runge-Kutta's, Milne's, *etc.* The simple predictor-corrector method given by Eqs. (29) and (31), which was also used in Report 3, will suffice in this case. In every jth iteration, instead of solving for i_{Ln} transcendentally [*cf.* Report 3, Eqs. (69) through (72), pp. 44-45], we first evaluate $(di_L/dt)_n$ using Eq. [44(c)], and then compute i_{Ln} from an expression analogous to Eq. (31), *i.e.*,

$$i_{Ln} = i_{L(n-1)} + 0.5\,\Delta t[(di_L/dt)_n + (di_L/dt)_{n-1}] \quad . \qquad (45)$$

Other than the above evaluation if i_{Ln}, the algorithm for computing the various variables if $L \neq 0$ is the same as in Report 3.

3. Noninductive Load

If $L = 0$, then the loop equation, Eq. [44(c)], becomes

$$N_c\dot{\phi}(t,\phi, i_L) - \frac{1}{C}q - Ri_L - E_k ln\left(1 + \frac{i_L}{I_0}\right) = 0 \quad . \qquad (46)$$

Regardless of whether a diode is present in the load or not, $dq/dt = i_L$ cannot be expressed explicitly as a function of t, ϕ, and q because $\dot{\phi}$ is an implicit function of i_L. Hence, we are unable to reduce Eqs. (44) into a set of two simultaneous differential equations with derivatives to the left of the equality signs and functions of the variables (t,ϕ, and q) only on the right side. For this reason, we are unable to use any of the conventional methods of numerical solution of differential equations directly. Instead, we substitute

$$q_n = q_{n-1} + 0.5\,\Delta t[i_{Ln} + i_{L(n-1)}] \tag{47}$$

into Eq. (46), obtain an implicit equation in i_{Ln}, and solve for i_{Ln} transcendentally. This explains the need for Newton's method in the algorithms described in Report 3. We could, of course, use other methods for the transcendental solution of i_{Ln}. We prefer, however, to use Newton's method because it is very simple and because it is characterized by quadratic convergence,[8] *i.e.*, the error at the jth iteration is proportional to the square of the error at the $(j-1)$th iteration (both errors are fractional if the condition for convergence is satisfied), thus resulting in a relatively fast convergence. The pitfall to watch for when using Newton's method is when f' [*cf.* Eqs. (58) and (60) in Report 3] becomes much smaller than unity.

4. Experimental and Computed $\dot{\phi}(t)$ and $i_L(t)$ of Inductively Loaded Core

Experimental and computed $\dot{\phi}(t)$ and $i_L(t)$ waveforms of a loaded core driven by step and ramp drives were compared in Figs. 15 and 16 of Report 3 (pp. 48 through 51). The tested core was Core J-1 (Lockheed 145SC1, 145-mil OD, 90-mil ID) whose switching parameters (*cf.* Report 3, p. 23) are as follows: l_i = 7.18 mm, l_o = 11.58 mm, ϕ_r = 31.0 maxwells, ϕ_s = 33.48 maxwells, H_a = 250 amp-turns/m, H_q = 26.0 amp-turns/m, H_n = 22.5 amp-turns/m, F_0'' = 0.27 amp-turn, ν = 1.43, λ = 1.64 ohm/turn$^{2.43}$amp$^{0.43}$, F_0 = 0.55 amp-turn, ρ_p = 2.27 ohms/turn^2, and F_B = 1.2 amp-turn. We have recomputed the $\dot{\phi}(t)$ and $i_L(t)$ waveforms for the cases in which $L \neq 0$, using the modified algorithm described above. The correspondingly modified computer program is given in Appendix F. The results were machine plotted and are compared in

Fig. 15 for step-F drive and in Fig. 16 for ramp-F drive with the same experimental oscillograms as in Report 3. It is quite clear that the agreement is better than in Figs. 15 and 16 of Report 3. Note, however, that the computed peak $\dot{\phi}$ is now lower than the experimental peak $\dot{\phi}$. This result is of no surprise. In fact, it was expected (but not found) in Report 3 (p. 47). Just as λ and ρ_p of ramp-F switching are smaller than step-F λ and ρ_p, one expects λ and ρ_p of switching under a monotonically *decreasing* $F(t)$ to be *larger* than step-F values. Such is the condition in the case of a loaded core switched by step-F or not-too-steep ramp-F drives. The more inductive the load is, the steeper is the decrease of $F(t)$ as ϕ varies from $-\phi_r$ to slightly above zero. This can be seen in Figs. 17 and 18 of Report 3 (pp. 52 and 53). Since step-F values of λ and ρ_p were used to compute the curves in Fig. 15, the resulting value of peak $\dot{\phi}$ is lower than the experimental value.

5. Summary

A modification is made in the algorithm given in Report 3 (p. 41-43) for computing $\dot{\phi}(t)$ and $i_L(t)$ of a loaded core switched by an arbitrary drive. If $L \neq 0$, conventional methods are used to solve three simultaneous differential equations of the first order. If $L = 0$, the loop equation is implicit in i_L, and Newton's method (or any other comparable method) must be applied in the numerical solution of i_L, ϕ, and $\dot{\phi}$. The corresponding computer program is given in Appendix F. The results for $L \neq 0$ are in a better agreement with experimental oscillograms than in Report 3.

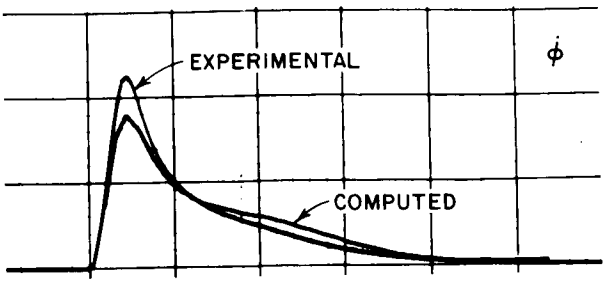
B. CORE-DIODE-TRANSISTOR BINARY COUNTER

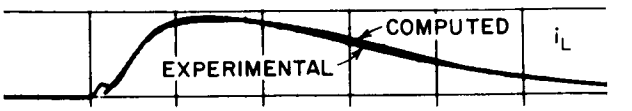
1. Operation

a. TWO-COUNT FLUX SWITCHING

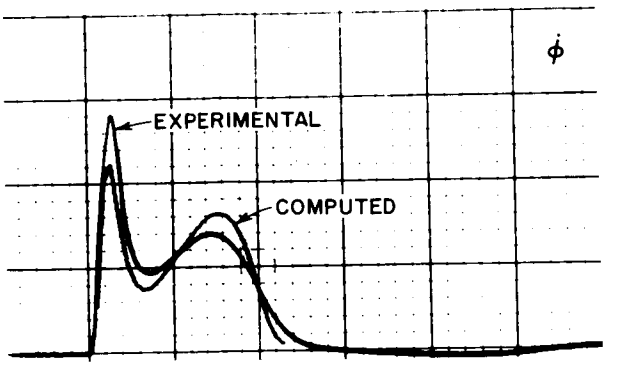
The circuit diagram for a single stage of a core-diode transistor binary counter is shown in Fig. 17.* Note that Cores 1′ and 2′ belong to the next stage. The circuit operation is described briefly as follows (circuit components and number of turns of windings used in this description are defined in Fig. 17): Initially, Cores 1 and 2 are in a CLEAR state. The counting operation is divided into four modes,

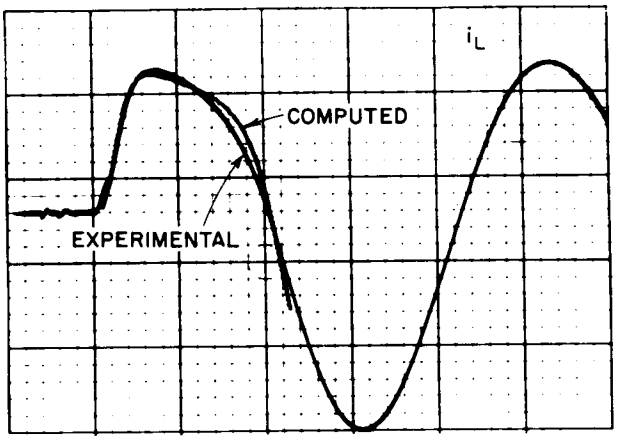
* This binary counter was developed by Alan J. DeVilbiss of Jet Propulsion Laboratory, Pasadena, California, for a future Mariner spacecraft.



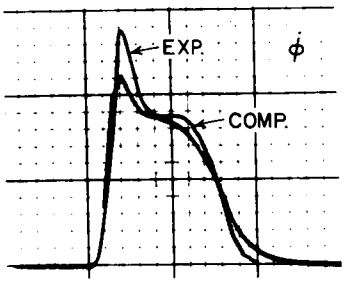


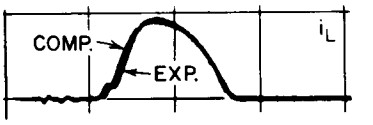
(a) $N_D I_D$ = 1.54 amp-turn;
N_c = 2; R_L = 1.58 Ω , L = 1.00 μhy





(b) $N_D I_D$ = 2.28 amp-turns;
N_c = 2; R_L = 0.131Ω, L = 0.38 μhy
C = 0.253 μf

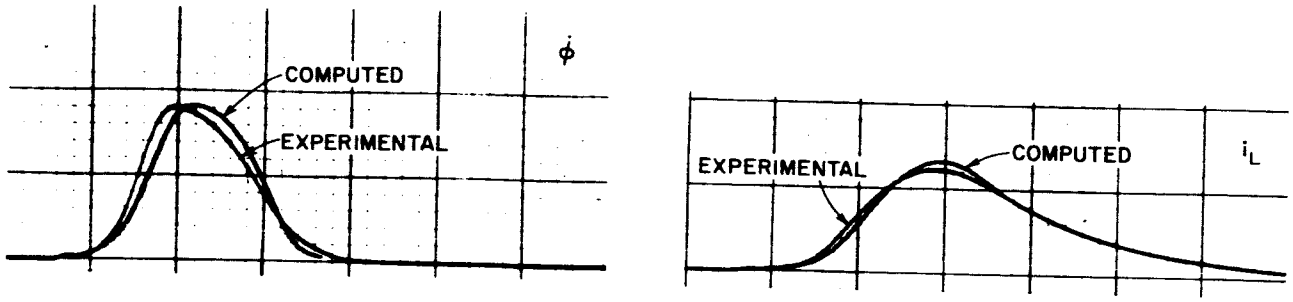




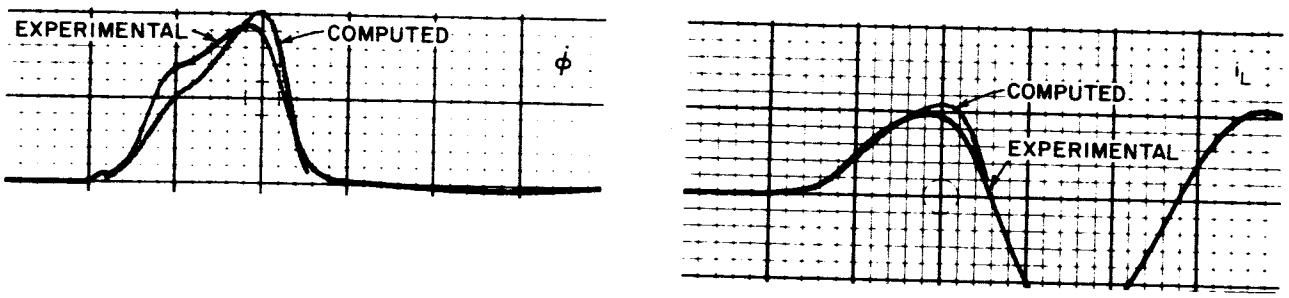
TC-5094-27

(c) $N_D I_D$ = 1.80 amp-turn;
N_c = 2; R_L = 0.131Ω , L = 0.38 μhy
C = 0.253 μf, 1N3604 Diode

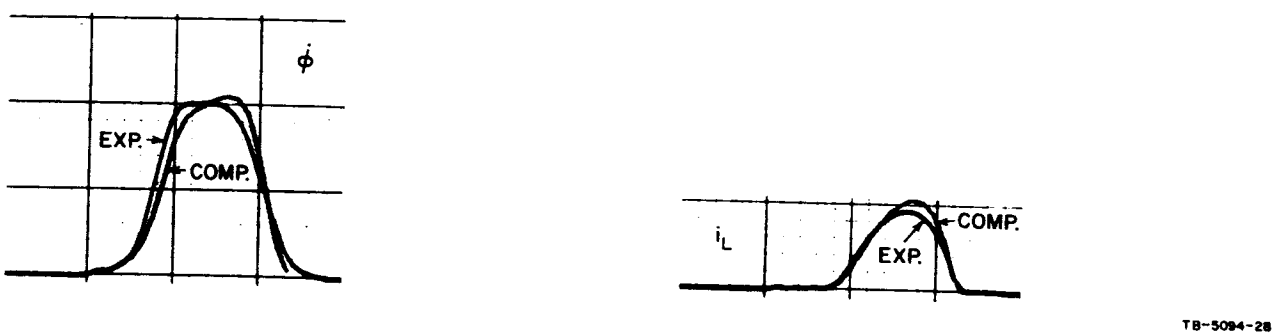
FIG. 15 EXPERIMENTAL AND COMPUTED $\dot{\phi}(t)$ AND $i_L(t)$ WAVEFORMS OF CORE J-1 WHICH IS INDUCTIVELY LOADED AND SWITCHED BY STEP-F DRIVE



(a) $N_D I_D$ = 3.40 amp-turns; T_r = 1.60 μsec
N_c = 2; R_L = 1.58Ω , L = 1.00 μhy



(b) $N_D I_D$ = 4.55 amp-turns; T_r = 1.40 μsec
N_c = 2; R_L = 0.131Ω , L = 0.38 μhy
C = 0.253 μf



(c) $N_D I_D$ = 3.40 amp-turns; T_r = 1.40 μsec
N_c = 2; R_L = 0.131Ω , L = 0.38 μhy
C = 0.253 μf, 1N3604 Diode

FIG. 16 EXPERIMENTAL AND COMPUTED $\dot{\phi}(t)$ AND $i_L(t)$ WAVEFORMS OF CORE J-1 WHICH IS INDUCTIVELY LOADED AND SWITCHED BY RAMP-F DRIVE

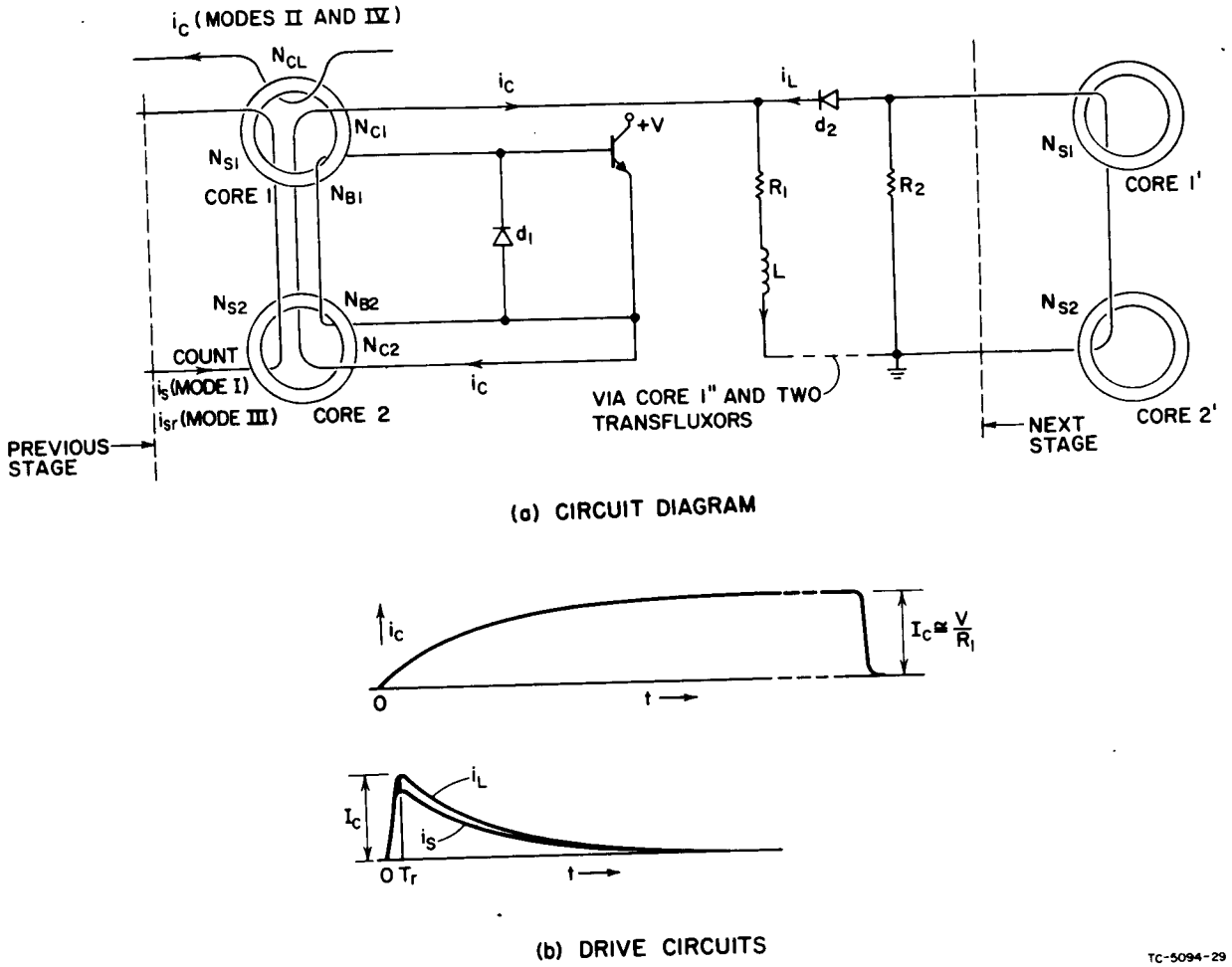


FIG. 17 A SINGLE STAGE OF A CORE-DIODE-TRANSISTOR BINARY COUNTER

where Modes I and II correspond to a STORE count, and Modes III and IV correspond to a CARRY count:

Mode I—An exponentially decaying current pulse, i_s, sets Cores 1 and 2 simultaneously. The flux switching is relatively fast. Diode d_1 is conducting, and the transistor is maintained in an OFF state.

Mode II—An exponentially rising current pulse i_c clears Core 1 at a relatively slow switching rate, while Core 2 remains in a SET state. Diode d_1 is conducting, and the transistor is maintained in the OFF state.

Mode III—An exponentially decaying current pulse, i_{sr}, tends to set both cores, but since Core 2 is already set, it only sets Core 1. The voltage $N_{B1}\dot{\phi}_1$ brings the transistor into the ON state, and the resulting collector current i_c helps current i_{sr} to set Core 1, thus maintaining

the transistor in the ON state (positive feedback, characteristic of blocking oscillator performance). Current i_c, as it builds up exponentially through an R-L circuit (diode d_2 is blocked), switches relatively slowly three additional cores that are not shown in Fig. 17: it clears (or tends to clear) Core 1", which is similar to Core 1, except two stages ahead (similarly, current i_c, which clears Core 1 in Mode II, is generated two stages behind); it clears a transfluxor core which is used for nondestructive readout of the binary state of the stage under discussion; and it sets the transfluxor core of the next stage. Toward the end of switching of Core 1, current i_{sr} has decayed to a negligible value, current i_c is approaching its asymptotic value, $I_c = V/R_1$ (to be exact, V stands for the supply voltage minus the collector-emitter voltage), and the transistor base current tends to decrease. Because of the decrease in i_{sr} and the base current and the increase in i_c, at a certain instant the net mmf of Core 2 becomes high enough to clear Core 2. As in the case of Core 1, a positive feedback maintains the transistor in the ON state until Core 2 reaches a CLEAR state. The switching of both Core 1 and Core 2 in Mode III is relatively slow because it is done sequentially. Upon termination of the switching in Core 2, the transistor turns off, Diode d_2 starts to conduct current i_L via R_1 and L, and the stored energy in the inductor L is dissipated in resistances R_1 and R_2 and in switching Core 1′ and Core 2′ (if the latter is initially not fully set) by current pulse i_s or in switching Core 1′ alone (if Core 2′ is initially fully set) by current pulse i_{sr}. Note that the current pulses i_s and i_{sr} driving the stage under discussion in Modes I and III, respectively, have been generated in a similar fashion in the previous stage.

Mode IV—A current pulse i_c clears Core 1 at a relatively slow switching rate, while Core 2 remains in a CLEAR state and the transistor is in the OFF state.

b. RANGE OF SUPPLY VOLTAGE

As the supply voltage, V, decreases below a certain value, the binary counter continues to function properly even though the cores are not fully set. A further decrease in V will cause a further decrease in $\Delta\phi_1$ and $\Delta\phi_2$, and it becomes necessary to determine how low $\Delta\phi_1$ and $\Delta\phi_2$ (or V) can be before the binary counter starts to malfunction. Let us examine incomplete flux switching during a cycle of two counts (four modes of operation), as shown in Fig. 18. In Mode I, switching is incomplete

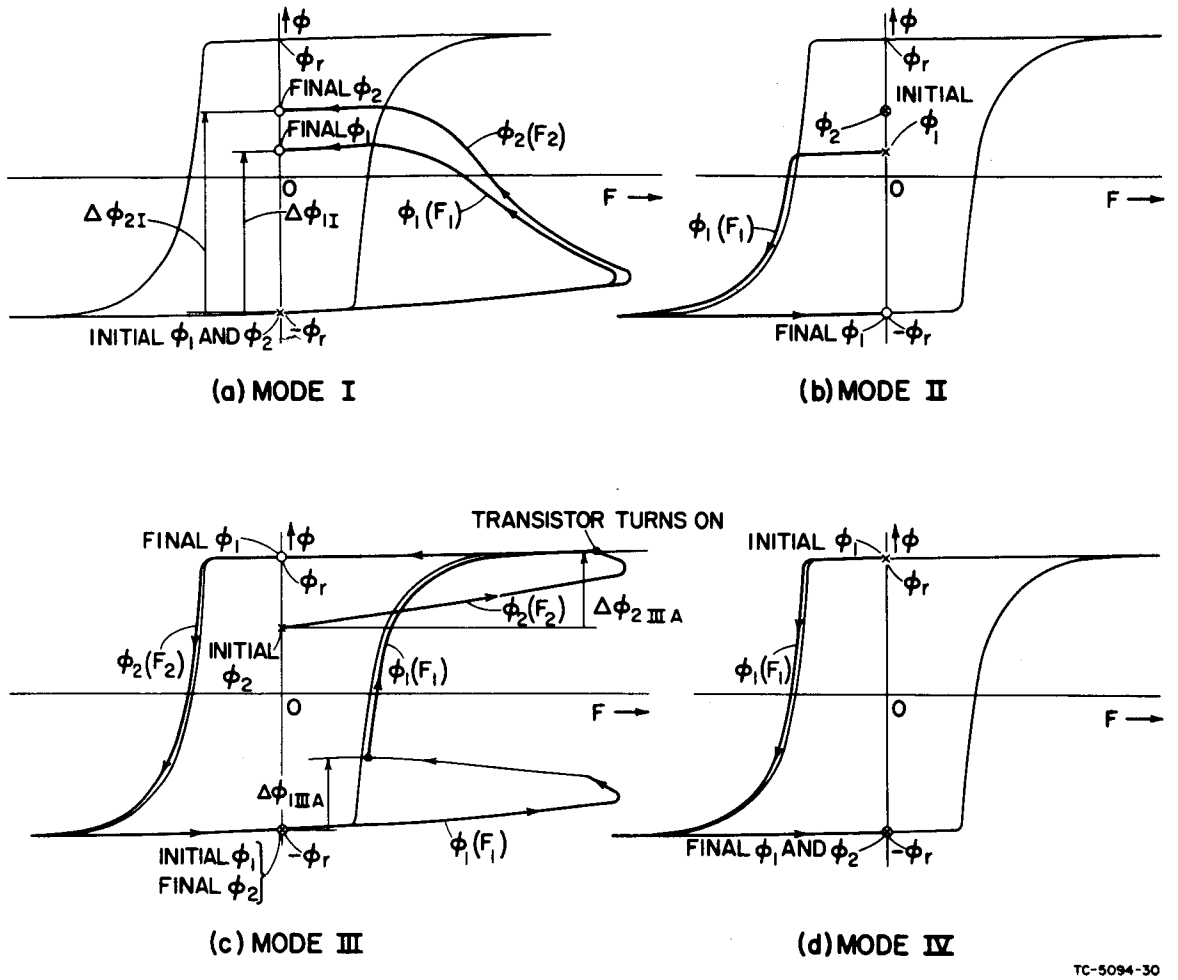


FIG. 18 INCOMPLETE FLUX SWITCHING IN FOUR MODES OF OPERATION OF A CORE-DIODE-TRANSISTOR BINARY COUNTER

because i_s either decays too fast or has a too-low amplitude or both (flux switching is mmf limited). In Mode II, Core 1 is cleared to $-\phi_r$ (current i_c is high for a long enough period to complete the required amount of switching). In the beginning of Mode III, to be referred to as Submode III-A, both cores start switching toward positive saturation as in Mode I, until Core 2 reaches positive saturation, and the transistor turns on. Core 1 completes its switching to $+\phi_r$, and Core 2 is then cleared to $-\phi_r$ because the transistor stays saturated as long as non-negligible flux switching takes place in at least one of the cores. In Mode IV, Core 1 is fully switched to $-\phi_r$ because i_c is high for a long enough period.

We see that the binary counter operates properly, even though only partial switching takes place in Mode I. How low can $\Delta\phi_1$ and $\Delta\phi_2$ be in Mode I? To answer this question, let us refer to Submode III-A. In order to turn the transistor on, we must assure that Core 2 stops switching before Core 1 stops switching, *i.e.*, that Core 2 reaches positive saturation before i_s decays to the stop-switching threshold value of Core 1. Let $\Delta\phi_{2\,\mathrm{I}}$ and $\Delta\phi_{2\,\mathrm{III\,A}}$ denote $\Delta\phi_2$ in Mode I and Submode III-A, respectively (ideally, $\Delta\phi_{2\,\mathrm{III\,A}} = 0$). If *both* $\Delta\phi_{2\,\mathrm{I}}$ and $\Delta\phi_{2\,\mathrm{III\,A}}$ are mmf-limited, and both are assumed to be governed by nearly the same switching parameters, then $\Delta\phi_{2\,\mathrm{I}} \cong \Delta\phi_{2\,\mathrm{III\,A}} \leq \phi_r$, because both result from the same insufficient drive current, i_s. Since we do not want $\Delta\phi_{2\,\mathrm{I}} + \Delta\phi_{2\,\mathrm{III\,A}}$ to be less than $2\phi_r$, we conclude that $\Delta\phi_{2\,\mathrm{I}}$ must exceed ϕ_r.

Let us examine $\Delta\phi_{2\,\mathrm{III\,A}}$ more carefully. In Submode III-A, Core 2 is initially in a partially switched state, and therefore both λ and F_0'' are smaller (*e.g.*, by 30%) than if initially $\phi_2 = -\phi_r$. These two factors are opposing: the decrease in λ requires a larger $\Delta\phi_{2\,\mathrm{I}}$, whereas the decrease in F_0'' allows $\Delta\phi_{2\,\mathrm{I}}$ to be smaller. Which factor predominates depends on the magnitude of the average net mmf of Core 2 during Submode III-A relative to F_0''. We must also remember that because of the "wing" in the static $\phi(F)$ curve, $\Delta\phi_{2\,\mathrm{I}} + \Delta\phi_{2\,\mathrm{III\,A}}$ may be slightly smaller than $2\phi_r$. It is thus quite clear that exact calculation of the lower limit of $\Delta\phi_{2\,\mathrm{I}}$ is extremely complex. As an approximation, we shall maintain our previous conclusion that in order to turn the transistor on in Mode III, we require that $\Delta\phi_{2\,\mathrm{I}} > \phi_r$. This criterion may be used to determine the minimum value of V, V_{min}.

Since the fast flux switching in Mode I determines the bottom boundary of the supply voltage, and since the slow flux switching in the other modes of operation can be analyzed manually by assuming that the *net* mmf follows the static $\phi(F)$ curve [*cf.* Figs. 18(b), (c), and (d)]—we shall proceed with the analysis of Mode I only.

2. Analysis of Mode I

a. Simplifying Assumptions

Before analyzing Mode I, let us refer to Fig. 17 and examine the termination of Mode III. When Core 2 is cleared to negative saturation and $\dot{\phi}_2$ approaches zero, the transistor starts to pull out of saturation because of insufficient base current. As a result, i_c starts to drop and a positive elastic $\dot{\phi}_2$ is generated. The voltage $N_{B2}\dot{\phi}_2$, which is clamped by Diode d_1, acts as a reverse bias on the transistor, and thus causes the latter to turn off faster than if the bias voltage were zero.

As soon as the drop in i_c is such that

$$-L\frac{di_c}{dt} = i_c R_1 \quad , \tag{48}$$

Diode d_2 starts to conduct current i_L. Equation (48) also holds true when the transistor is completely turned off. The variation of i_c during the transient period in which the transistor is pulled out of saturation depends on the base current and the collector current. A solution for i_c during this period is complex and is beyond the scope of this work. We shall, therefore, simplify the problem by assuming that during the above transient period, $i_L(t)$, as it rises from zero to I_c (the current value that L tends to maintain), is given and that Eq.(48) holds, *i.e.*, the voltage across R_1 and L is

$$v_{R_1\text{-}L} = R_1 i_L + L\frac{di_L}{dt} \quad . \tag{49}$$

b. Basic Equations

Mode I begins as soon as the transistor of the *previous* stage starts to pull out of saturation. Based on the simplifying assumptions above and since the forward voltage drop across Diode d_2 is much smaller

than $R_1 i_L + L(di_L/dt)$, the equivalent circuit for Mode I is as shown in Fig. 19. The total resistance of the windings with N_{s1} and N_{s2} turns is denoted by R_3 and that of the windings with N_{B1} and N_{B2} turns is denoted by R_4.

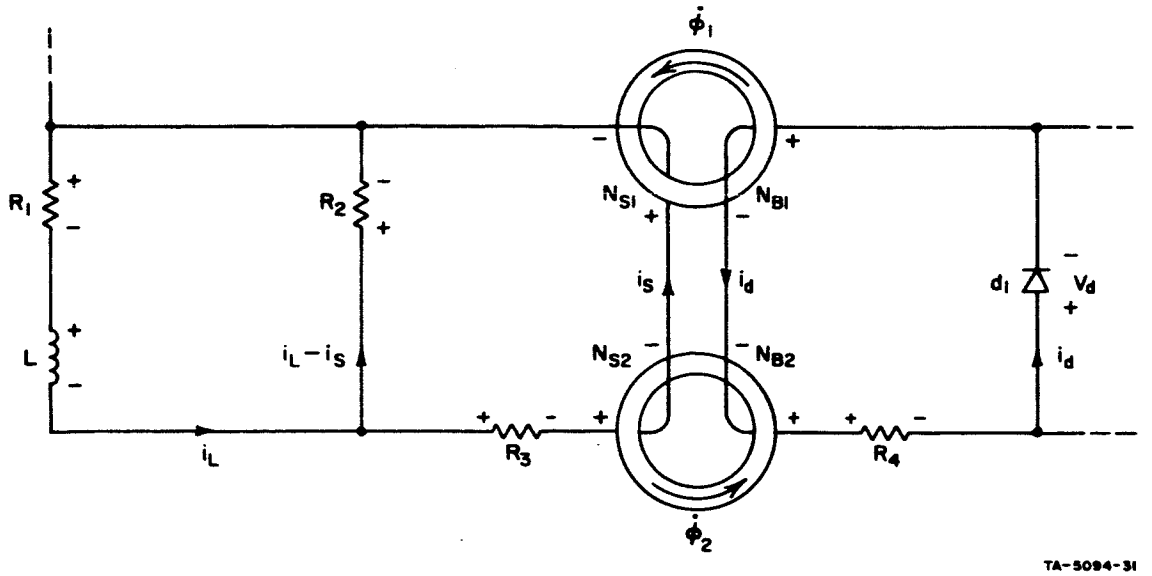


FIG. 19 EQUIVALENT CIRCUIT FOR MODE I OF A SINGLE STAGE IN A CORE-DIODE-TRANSISTOR BINARY COUNTER

There are seven equations with seven unknowns. The seven equations include two mmf equations (one for each core), two $\dot{\phi}(F,\phi)$ equations (one for each core), and three loop equations. The unknowns are the following time variables: i_L (beyond its rise time), i_s (the current driving the cores), i_d (the load current in Diode d_1), F_1 (the net mmf of Core 1), F_2 (the net mmf of Core 2), ϕ_1 (the flux of Core 1), and ϕ_2 (the flux of Core 2).

By inspection of Fig. 19, the mmf equations are

$$F_1 = N_{s1} i_s + N_{B1} i_d \tag{50}$$

and

$$F_2 = N_{s2} i_s - N_{B2} i_d \quad . \tag{51}$$

Using Eqs. (2) and (18) through (20),

$$\dot{\phi}_1 = \epsilon \dot{F}_1 + \dot{\phi}(F_1, \phi_1) \tag{52}$$

and

$$\dot{\phi}_2 = \epsilon \dot{F}_2 + \dot{\phi}(F_2, \phi_2) \quad , \tag{53}$$

where $\dot{\phi}(F,\phi) = \dot{\phi}_p\{1 - [(2\phi + \phi_s - \phi_d)/(\phi_s + \phi_d)]^2\}$. Note that $\dot{\phi}(F,\phi)$ is used as an approximation for $\dot{\phi}_{ma}(F,\phi) + \dot{\phi}_i(F,t,T_i)$, Eq. (22). By inspection of Fig. 19, the three loop equations are

$$L\frac{di_L}{dt} + i_L R_1 + i_s R_3 + N_{s1}\dot{\phi}_1 + N_{s2}\dot{\phi}_2 = 0 \tag{54}$$

$$i_s R_3 + N_{s1}\dot{\phi}_1 + N_{s2}\dot{\phi}_2 - (i_L - i_s)R_2 = 0 \tag{55}$$

and

$$i_d R_4 + V_d + N_{B1}\dot{\phi}_1 - N_{B2}\dot{\phi}_2 = 0 \quad , \tag{56(a)}$$

where [*cf.* Eq. (42)]

$$V_d = i_d R_d + E_k \ln\left(1 + \frac{i_d}{I_0}\right) \tag{56(b)}$$

is the forward voltage drop across Diode d_1.

We now wish to apply the models for inelastic and elastic switching in the numerical solutions of Eqs. (50) through (56).

3. Transcendental Solution of Currents

We wish to solve Eqs. (50) through (56) at every nth Δt. For the sake of convenience, the subscript n will be omitted entirely and the subscripts $(n\text{-}1)$, $(n\text{-}2)$, *etc.*, will be replaced by the subscripts (-1), (-2), *etc.* In other words, at every nth Δt, we identify n with zero.

Since the functions F_1 *vs.* t and F_2 *vs.* t are not known *a priori*, we shall evaluate $\dot{F}_1$ and $\dot{F}_2$ as ratios of differences rather than differentials, *i.e.*,

$$\dot{F}_1 = [F_1 - F_{1(-1)}]/\Delta t \tag{57}$$

and

$$\dot{F}_2 = [F_2 - F_{2(-1)}]/\Delta t \quad . \tag{58}$$

Note that $F_{1(-1)}$ and $F_{2(-1)}$ are known quantities which were evaluated while solving for the time variables of the previous Δt. Substitution of Eqs. (50) and (57) into Eq. (52) and substitution of Eqs. (51) and Eq. (58) into Eq. (53) results in two simultaneous nonlinear differential equations which are formally expressed as

$$\dot{\Phi}_1 = f_1(i_s, i_d, \Phi_1) \tag{59}$$

and

$$\dot{\Phi}_2 = f_2(i_s, i_d, \Phi_2) \quad . \tag{60}$$

Substitution of Eqs. (59) and (60) into Eq. (54) gives

$$\frac{di_L}{dt} = -\frac{1}{L}\left[i_L R_1 + i_s R_3 + N_{s1} f_1(i_s, i_d, \Phi_1) + N_{s2} f_2(i_s, i_d, \Phi_2)\right] \quad . \tag{61}$$

Note that Eqs. (59), (60), and (61) form a set of three simultaneous differential equations of the first order.

If Eqs. (55) and (56) did not include $\dot{\Phi}_1$ and $\dot{\Phi}_2$, they could be solved for i_s and i_d. Knowing i_s and i_d, we could then use conventional methods in order to solve Eqs. (59) through (61), which form a set of three simultaneous differential equations with differentials on the left side and functions of unknown time variables (Φ_1, Φ_2, and i_L) on the right side of the equality signs. However, since Eqs. (55) and (56) do include $\dot{\Phi}_1$ and $\dot{\Phi}_2$, we have to modify the conventional methods of solving differential equations by incorporating a transcendental solution for i_s and i_d. The latter will be done next, using Newton's method for solving two simultaneous implicit equations.[9]

Following Eqs. (56) and (55), we are looking for the roots i_d and i_s of the equations

$$f = N_{B2}\dot{\Phi}_2 - N_{B1}\dot{\Phi}_1 - i_d(R_d + R_4) - E_k \ln\left(1 + \frac{i_d}{I_0}\right) = 0 \tag{62}$$

and

$$g = n_{s2}\dot{\phi}_2 + N_{s1}\dot{\phi}_1 + i_s(R_2 + R_3) - i_L R_2 = 0 \quad . \tag{63}$$

Following Eqs. (50) and (51), $\partial F_1/\partial i_d = N_{B1}$, $\partial F_2/\partial i_d = -N_{B2}$, $\partial F_1/\partial i_s = N_{s1}$, and $\partial F_2/\partial i_s = N_{s2}$. Hence, the partial derivatives of f and g are as follows:

$$\frac{\partial f}{\partial i_d} = -\left(N_{B2}^2\dot{\phi}_2' + N_{B1}^2\dot{\phi}_1' + R_d + R_4 + \frac{E_k}{i_d + I_0}\right) \quad , \tag{64}$$

$$\frac{\partial g}{\partial i_s} = N_{s2}^2\dot{\phi}_2' + N_{s1}^2\dot{\phi}_1' + R_2 + R_3 \quad , \tag{65}$$

and

$$\frac{\partial f}{\partial i_s} = -\frac{\partial g}{\partial i_d} = N_{s2}N_{B2}\dot{\phi}_2' - N_{s1}N_{B1}\dot{\phi}_1' \quad . \tag{66}$$

The corrections to be added to i_d and i_s at each iteration (until both are negligible) are[9]

$$\delta i_d = \frac{1}{D}\left(-f\frac{\partial g}{\partial i_s} + g\frac{\partial f}{\partial i_s}\right) \tag{67}$$

and

$$\delta i_s = \frac{1}{D}\left(f\frac{\partial g}{\partial i_d} - g\frac{\partial f}{\partial i_d}\right) \tag{68}$$

where

$$D = \left(\frac{\partial f}{\partial i_d}\right)\left(\frac{\partial g}{\partial i_s}\right) - \left(\frac{\partial f}{\partial i_s}\right)\left(\frac{\partial g}{\partial i_d}\right) \quad . \tag{69}$$

Under certain conditions, the convergence to the correct solution of i_d (or i_s) is *oscillatory* and *slow*. This can be detected by noticing that δi_d (or δi_s) alternates sign between two consecutive iterations and that its absolute value decreases by only a small percentage. In many cases, this situation may be remedied by correcting i_d (or i_s) by *half* of δi_d (or δi_s), as was done in Report 3 [Eq. (58a), p. 42].

The transcendental solution of i_d and i_s described above will be incorporated into the numerical solution of the time variables, using two types of numerical solutions of differential equations, a simple method and a more complex one (Runge-Kutta's followed by Adams'). This duplication is designed in order to check the accuracy of using a simple method in this type of problem by comparing the results with the ones obtained by using a more complex and exact method.

4. Computation Using a Simple Method

a. Method of Solution

Simple predictor-corrector equations, similar to Eqs. (29) and (31), are used. Letting y stand for ϕ_1, ϕ_2, and i_L, these are:

$$y = y_{(-2)} + 2\Delta t \dot{y}_{(-1)} \tag{70}$$

for prediction, and, after $\dot{y}$ is evaluated from the differential equation,

$$y = y_{(-1)} + 0.5\,\Delta t\,[\dot{y} + \dot{y}_{(-1)}] \tag{71}$$

for correction.

b. Outline for Computer Program

The computer program for the core-diode-transistor binary counter, using a simple method of solution, is given in Appendix G. The program is written in ALGOL-60 language, and has the following outline:

(1) Declare all identifiers (core parameters, circuit parameters, variables, and miscellaneous), output lists and formats.

(2) Declare PROCEDUREs $\dot{\phi}_\epsilon(F, \Delta t, NV, \dot{\phi}'_\epsilon)$, Appendix A, and $\dot{\phi}(F, \phi, \dot{\phi}, \dot{\phi}^*)$, Appendix D-2.

(3) Set values of core and circuit parameters.

(4) Set initial values of variables.

(5) Compute a rough approximation for the value of τ_s, using the relations*

$$I_s = \left(\frac{V}{R_1}\right) \frac{R_2}{R_2 + 0.6\ \rho_p (N_{s1}^2 + N_{s2}^2)} \tag{72}$$

and

$$\tau_s \cong \frac{2\phi_r (N_{B1}^2 + N_{B2}^2)}{0.6\ N_{B2}\rho_p [I_s (N_{s1}N_{B1} + N_{s2}N_{B2}) - F_0''(N_{B1} + N_{B2})]} \tag{73}$$

Compute $I_c = V/R_1$.

(6) Print heading (core parameters, circuit parameters, and output variables) and initial values of output variables.

(7) For every nth Δt during switching time, do the following:

(a) Lower the Δt index of the time variables computed previously by one.

(b) Set $\Delta t = \tau_s/1000$ during the rise time, T_r, of $i_L(t)$ and $\Delta t = \tau_s/500$ for $t \geq T_r$.

(c) If $t \leq T_r$, compute i_L from a given empirical expression.

(d) Compute *first approximations* for i_L (if $t > T_r$), ϕ_1, and ϕ_2, using Eq. (70), and for i_s and i_d, using the relations $i_s \cong 2i_{s(-1)} - i_{s(-2)}$ and $i_d \cong 2i_{d(-1)} - i_{d(-2)}$.

(e) Until a specified convergence condition is achieved, compute the following steps in a *loop*:

F_1 [Eq. (50)];
F_2 [Eq. (51)];
$\dot{F}_1$ [Eq. (57)];
$\dot{F}_2$ [Eq. (58)];
$\dot{\phi}_{ma1} = \dot{\phi}(F_1, \phi_1, \dot{\phi}'_{ma1}, \dot{\phi}_1^*)$ [Appendix D-2];
$\dot{\phi}_{\epsilon 1} = \dot{\phi}_\epsilon(\dot{F}_1, \Delta t, NV, \dot{\phi}'_{\epsilon 1})$ [Appendix A];
$\dot{\phi}_1 = \dot{\phi}_{ma1} + \dot{\phi}_{\epsilon 1}$;
$\dot{\phi}'_1 = \dot{\phi}'_{ma1} + \dot{\phi}'_{\epsilon 1}$;

---

* Equations (72) and (73) have been derived by using average, constant values of ρ, $\overline{\rho} \cong 0.6\ \rho_p$, in a rough calculation of the switching time for $\Delta\phi_1 = 2\phi_r$.

$\dot{\phi}'_{ma2} = \dot{\phi}(F_2, \phi_2, \dot{\phi}'_{ma2}, \dot{\phi}^*_2)$ [Appendix I)-2];
$\dot{\phi}_{\epsilon 2} = \dot{\phi}_{\epsilon}(\dot{F}_2, \Delta t, NV, \dot{\phi}'_{\epsilon 2})$ [Appendix A];
$\dot{\phi}_2 = \dot{\phi}_{ma2} + \dot{\phi}_{\epsilon 2}$;
$\dot{\phi}'_2 = \dot{\phi}'_{ma2} + \dot{\phi}'_{\epsilon 2}$;
If $t > T_r$, di_L/dt [Eq. (54)] and i_L [Eq. (71)];
ϕ_1 [Eq. (71)];
ϕ_2 [Eq. (71)];
V_d [Eq. (56b) if $N_{B2}\dot{\phi}_2 > N_{B1}\dot{\phi}_1$; otherwise, $N_{B2}\dot{\phi}_2 - N_{B1}\dot{\phi}_1$];
f [Eq. (62)];
g [Eq. (63)];
$\partial f/\partial i_d$ [Eq. (64)];
$\partial g/\partial i_s$ [Eq. (65)];
$\partial f/\partial i_s$ [Eq. (66)];
$\partial g/\partial i_d = -\partial f/\partial i_s$;
D [Eq. (69)];
δi_d [Eq. (67)];
δi_s [Eq. (68)].

Add δi_d to previous i_d, but if the sign of δi_d is different from the previous one, add only $0.5\ \delta i_d$; add δi_s to previous i_s, but if the sign of δi_s is different from the previous one, add only $0.5\ \delta i_s$.

Repeat the above steps if either $|\delta i_d| > 0.0001\ |i_d|$ or $|\delta i_s| > 0.0001\ |i_s|$, but no more than 19 times. Count the number of times this convergence condition fails to be satisfied.

Repeat Steps (a) through (e) unless $\dot{\phi}_1 = 0$ *and* $\dot{\phi}_2 = 0$ *and* $\phi_1 > -0.9\ \phi_r$.

(8) Print output (t, i_L, i_s, $F_1, \phi_1, \dot{\phi}_1, F_2, \phi_2\ \dot{\phi}_2, i_d, V_d$, $\dot{\phi}_{\epsilon 1}, \dot{\phi}_{\epsilon 2}$, number of iterations, cumulative number of convergence failures), say, once every 20th Δt.

c. EXPERIMENTAL VERIFICATION

A single stage of a core-diode-transistor binary counter, Fig. 17, was built using the following components:

Core

Lockheed 100SC1 ferrite core (OD = 100 mils, ID = 70 mils, h = 30 mils) whose switching parameters at room temperature (25°C) are as follows: l_i = 5.59 mm; l_o = 7.98 mm; ϕ_r = 6.25 maxwells; ϕ_s = 7.00 maxwells; H_a = 290 amp-turns/m; H_q = 42.7 amp-turns/m; H_n = 38.0 amp-turns/m; ν = 1.207; λ = 0.64 ohm/turn$^{2.207}$amp$^{0.207}$; F_0'' = 0.35 amp-turn; ρ_p = 0.948 ohm/turn^2; F_0 = 0.805 amp-turn; F_B = 3.0 amp-turn.

Circuit

Number of turns: N_{s1} = 11; N_{s2} = 12; N_{B1} = 16; N_{B2} = 20; $N_{c1} = N_{c2}$ = 12.

Resistances (ohms): R_1 = 107.36; R_2 = 199.55; R_3 = 0.34; R_4 = 0.53.

Inductance: L = 0.202 millihenry.

Diodes d_1 and d_2: FD 643; E_k = 0.0578 volt; I_0 = 0.0615 microampere; R_d = 0.1 ohm.

Transistor: 2N1613.

Drive

Supply voltage (minus collector-emitter voltage drop): V = 27 volts; 8.6 volts.

Rise time of i_L: T_r = 0.13 microsecond. Using a library computer program for polynomial curve fitting, the following empirical expressions for $i_L(t)$ during $0 \le t \le T_r$ were fit to the rising portion of $i_L(t)$ waveform (V = 27 volts) from i_L = 0 (at t = 0) to i_L = 0.252 ampere (at t = 0.13 μsec), in which $I_c = V/R_1$, t is in seconds, and $T_n = 13 - t \cdot 10^8$:

$$i_L = \frac{I_c}{0.252} \cdot \begin{cases} 1.02 \cdot 10^{19} t^{8/3} & \text{if } 0 \le t < 0.02 \cdot 10^{-6} \\ 4 \cdot 10^6 t - 0.05 & \text{if } 0.02 \cdot 10^{-6} \le t < 0.06 \cdot 10^{-6} \\ 0.252 - 0.01\big(0.00947 + T_n\{-0.3169 + T_n[1.729261 & \\ \quad + T_n(-0.575947 + T_n \cdot 0.073769)]\}\big) & \text{if } 0.06 \cdot 10^{-6} \le t < 0.13 \cdot 10^{-6} \end{cases} \tag{74}$$

The above core, circuit, and drive parameters were inserted into the computer program, and the results were machine plotted. Experimental oscillograms of $i_L(t)$, $i_s(t)$, $\dot{\phi}_1(t)$, $\dot{\phi}_2(t)$, $i_d(t)$, $V_d(t)$ were superimposed on computed waveforms. These are compared in Fig. 20 for V = 27.0 volts and in Fig. 21 for V = 8.6 volts.

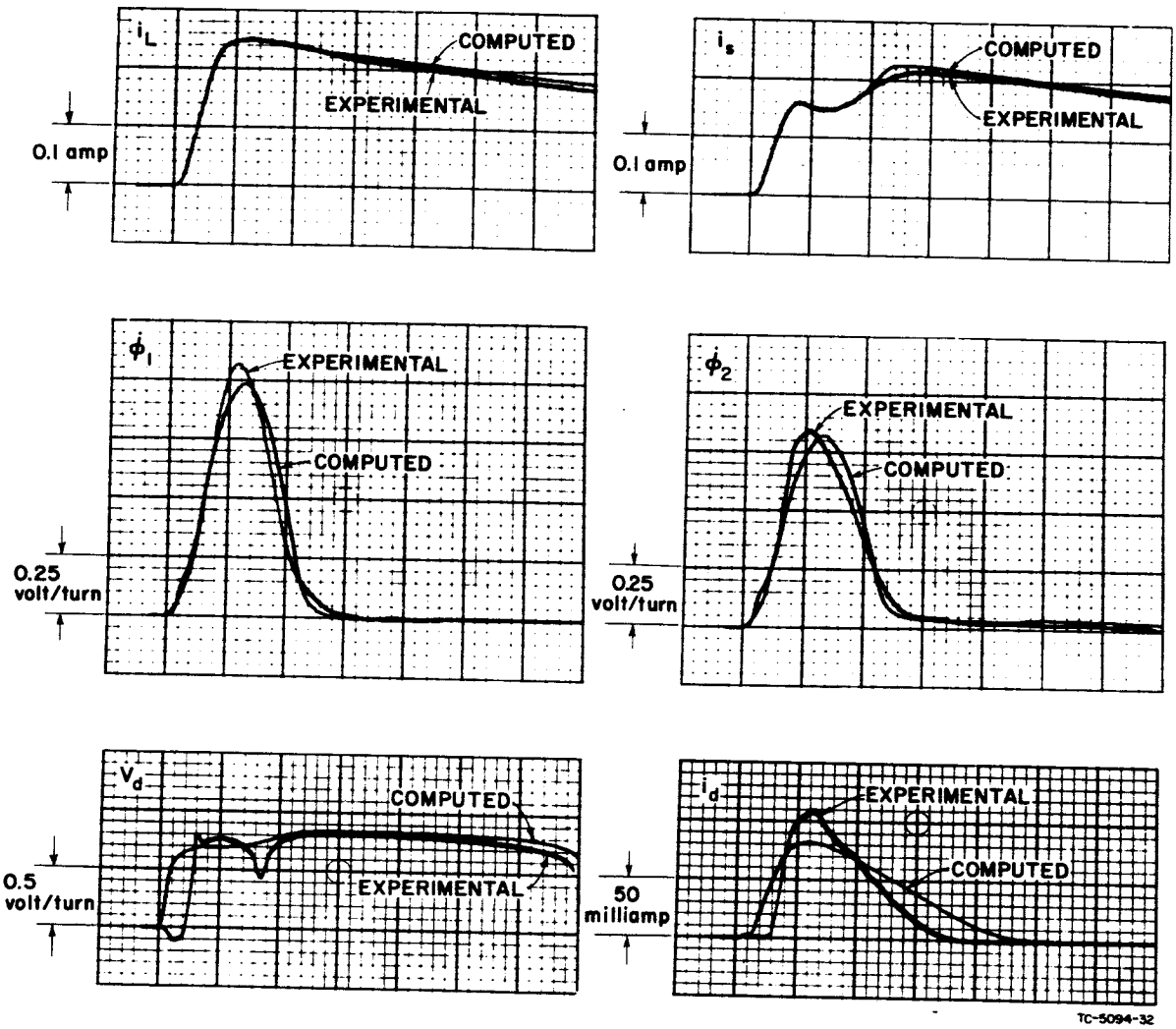


FIG. 20 EXPERIMENTAL AND COMPUTED CURRENT AND VOLTAGE WAVEFORMS IN MODE I OF A CORE-DIODE-TRANSISTOR BINARY COUNTER FOR V = 27 VOLTS
Time scale = 0.1 μsec/major div.

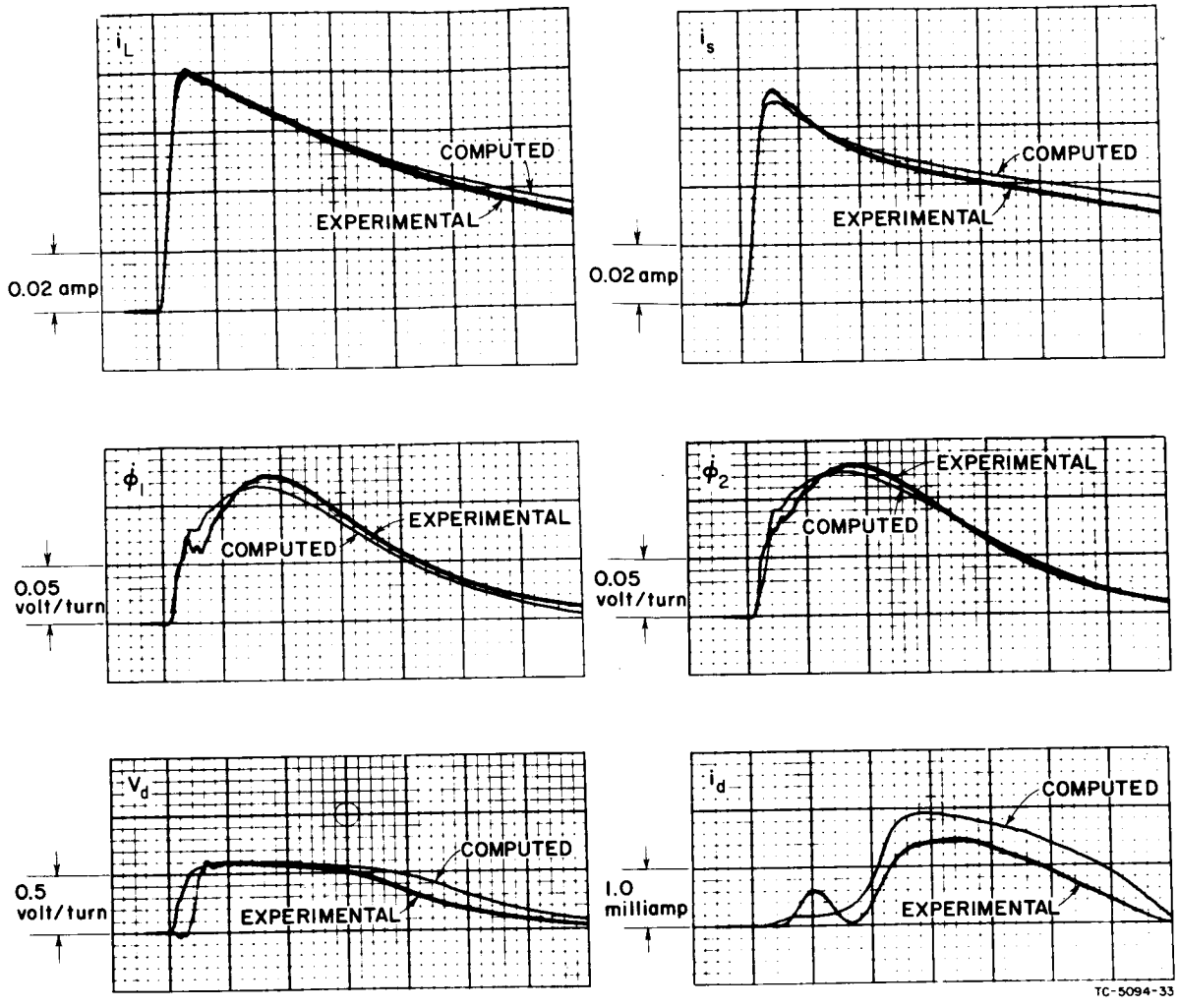


FIG. 21 EXPERIMENTAL AND COMPUTED CURRENT AND VOLTAGE WAVEFORMS IN MODE I OF A CORE-DIODE-TRANSISTOR BINARY COUNTER FOR V = 8.6 VOLTS
Time scale = 0.2 μsec/major div.

5. Computation Using Runge-Kutta and Adams Methods

a. Method of Solution

The computation described in the previous section is based on a simple method of solving a set of differential equations. One may, then, doubt the accuracy of the computation. In order to check this accuracy, a second computer program was written, using conventional and more complex methods of solution: the *Runge-Kutta* method for starting the computation in the first four Δt, and the *Adams* method thereafter. The description of these methods may be found in several textbooks.[7,9]

b. COMPUTER PROGRAM

The computer program for the core-diode-transistor binary counter, using the Runge-Kutta and Adams methods, is given in Appendix H. The program language is ALGOL-60. The outline of this program is similar to the outline of the program using simple methods of solution, Part B-4-b, except for the following. The differential equations and the transcendental solutions for i_d and i_s are declared in a special PROCEDURE, *F(X, Y, DX)* in which *X* stands for the independent variable (t), *Y* stands for an array of the dependent variables (ϕ_1, ϕ_2, and i_L), and *DX* stands for an array of the derivatives ($\dot{\phi}_1$, $\dot{\phi}_2$, and di_L/dt).

The computer program for the Runge-Kutta method is a slightly modified version of a library PROCEDURE RKSTARTS (*K, NF, X1, H, Y, YPR, F*),[10] where *K* is the number of differential equations (here, $K = 3$), *NF* is the number of Δt steps for which the computation is repeated (with the use of Adams method, $NF = 4$), *X1* is the initial value of the independent variable (here, $X1 = t_0 = 0$), *H* is the step size (here, $H = \Delta t$), *Y* and *YPR* are identifiers for two $(NF + 1) \cdot K$ arrays for storing the values of the dependent variables and their derivatives, respectively, (at $t = 0, \Delta t, 2\Delta t, \ldots, NF \cdot \Delta t$), and *F* is the PROCEDURE *F(X, Y, DX)*. The computer program for the Adams method is also a slightly modified version of a library PROCEDURE,[11] ADAMS (*X, Y, YPRIME, N, EU, EL, EPS, H, HMIN*), where *X* is the independent variable (t), *Y* and *YPRIME* are arrays similar to those used above for the dependent variables and their derivatives, *N* is the number of differential equations (here, $N = 3$), *EU* and *EL* are upper and lower bounds, respectively, of relative error, *EPS* is the minimum absolute value for any dependent variable to be used as a reference for evaluation of the relative error, *H* is the step size (Δt) that depends on *EU* and *EL* (if *any* equation error exceeds *EU*, *H* is halved, and if *all* errors are below *EL*, *H* is doubled), and *HMIN* is the minimum value that *H* may be reduced to. It can be seen in Appendix H that $\Delta t = \tau_s/500$, where τ_s is evaluated from Eqs. (72) and (73), and $\min(\Delta t) = \Delta t/10$. Also, during $0 \leq t \leq T_r$, $EU = 0.001$ and $EL = 0.0001$, whereas if $t > T_r$, $EU = 0.005$ and $EL = 0.001$. For $t > T_r$, di_L/dt was solved for, using Eq. (54), but for $t \leq T_r$, di_L/dt was derived by differentiating Eq. (74):

$$\frac{di_L}{dt} = \frac{I_c}{0.252} \cdot \begin{cases} 2.72 \cdot 10^{19} t^{5/3} & \text{if} \quad 0 \le t < 0.02 \cdot 10^{-6} \\ 4 \cdot 10^{6} & \text{if} \quad 0.02 \cdot 10^{-6} \le t < 0.06 \cdot 10^{-6} \\ 10^{5}\{-0.316910 + T_n[3.458522 & \\ + T_n(-1.727841 + T_n \cdot 0.295076)]\} & \text{if} \quad 0.06 \cdot 10^{-6} \le t < 0.13 \cdot 10^{-6} \end{cases} \tag{75}$$

where t is measured in seconds and $T_n = 13 - t \cdot 10^8$.

c. RESULTS

The computed results came out very close to those obtained by the simple method of solution. The largest differences were in the third place after the decimal point, and could hardly be detected in the machine plots of the results. This is demonstrated in Fig. 22 by superimposing computed $\dot{\phi}_1(t)$ and $\dot{\phi}_2(t)$ waveforms for $V = 27$ and $V = 8.6$ volts obtained by the two methods of solution. The agreement between any other pair of computed waveforms is either similar to or better than that shown in Fig. 22.

6. Discussion

a. VALIDITY OF COMPUTATION

The agreement between the computed and the experimental results is satisfactory in general. In the case of $i_d(t)$, the agreement is poor because i_d is a nonlinear function of the voltage across the diode, V_d, and because V_d is the difference between two voltages, $N_{B2}\dot{\phi}_2$ and $N_{B1}\dot{\phi}_1$, whose magnitudes are close to each other. Any small relative error in $\dot{\phi}_2$ or $\dot{\phi}_1$ or both (if of opposite signs) will result in a magnified relative error in i_d if V_d is above the "bias" voltage, *i.e.*, above the knee of the forward characteristic of the diode.

The computation of the time variables during the rise time of $i_L(t)$ is limited by two drawbacks: First, the collector current, i_c, is ignored by assuming that $L(di_c/dt) + i_c R_1 = 0$, Eq. (48), while the transistor pulls out of saturation. Second, empirical expressions for $i_L(t)$ are fit to experimental data. Both drawbacks could be overcome by incorporating a dynamic switching model for the transistor. This was not done because of the added complexity resulting from the use of this model and

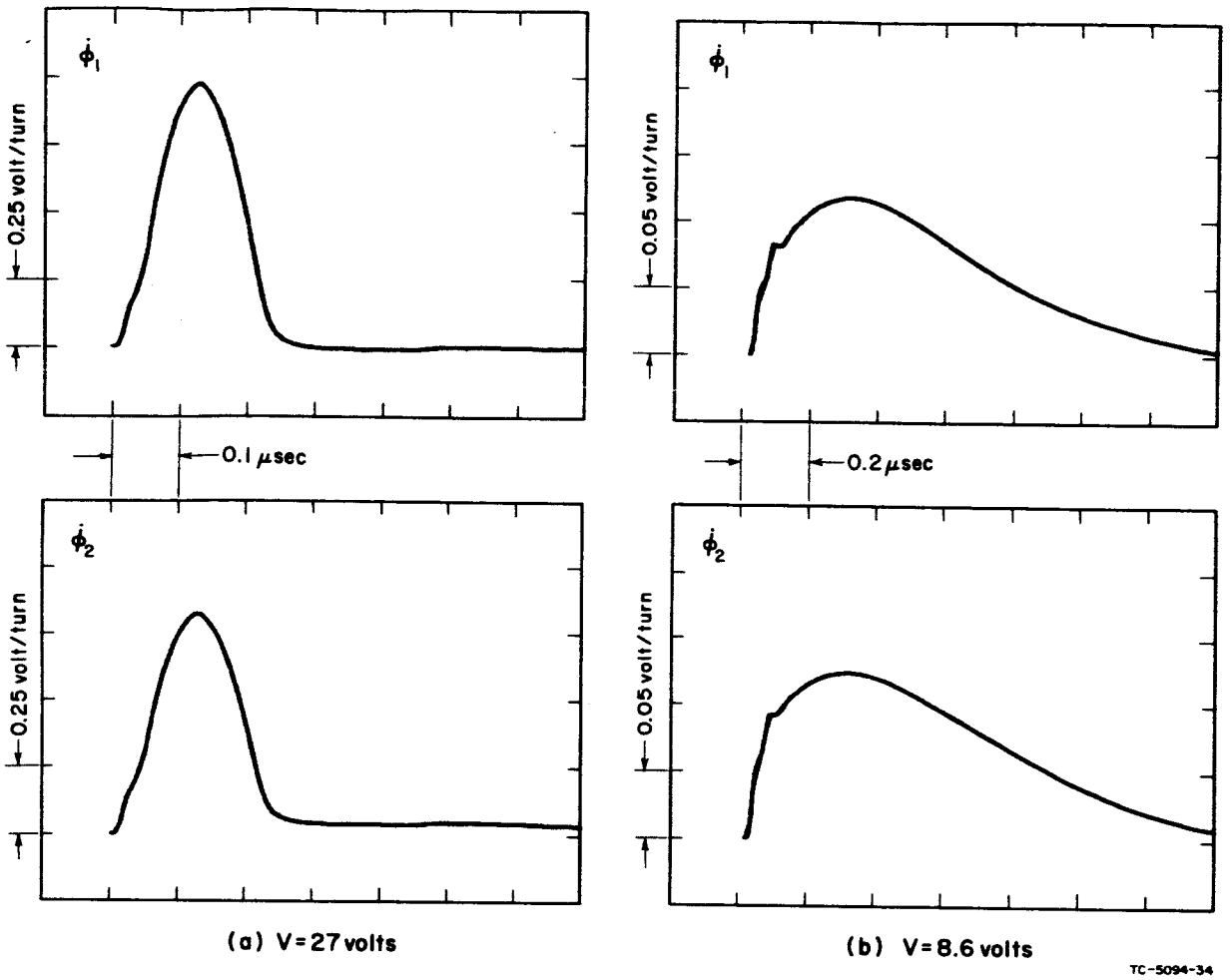


FIG. 22 SUPERPOSITION OF $\dot{\phi}(t)$ WAVEFORMS COMPUTED BY A SIMPLE METHOD AND $\dot{\phi}(t)$ WAVEFORMS COMPUTED BY RUNGE-KUTTA AND ADAMS METHODS

because such a model is beyond the scope of our work. Instead of fitting an exact function for $i_L(t)$, we could have assumed a sinusoidal function or a ramp function between $t = 0$ and $t = T_r$. Although such alternatives do not depend on curve fitting, the rise time still has to be furnished and the results of computation include larger errors. Another alternative with an even larger error is to ignore the switching time of the transistor and assume that $T_r = 0$.

b. VARIATIONS OF $\phi_1(F_1)$ and $\phi_2(F_2)$ DURING SWITCHING TIME

Having computed F_1, ϕ_1, F_2, and ϕ_2 *vs.* t, we are able to determine the variations of ϕ_1 *vs.* F_1 and ϕ_2 *vs.* F_2 during the switching time by treating t as a parameter. These variations were machine plotted for the cases of V = 27 volts and V = 8.6 volts, and are shown superimposed on manually plotted static $\phi(F)$ curves in Fig. 23.

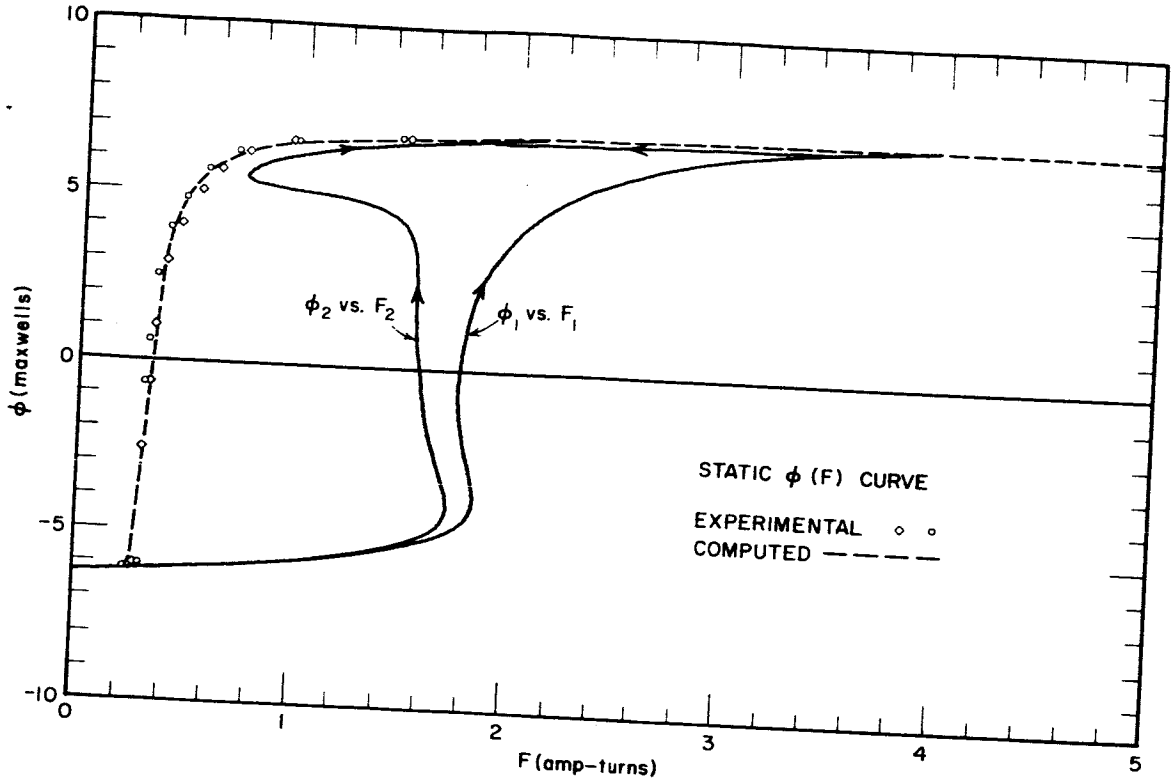
It can be seen in Fig. 23 that the flux switching is complete for V = 27 volts, but for V = 8.6 volts, the flux switching is only partial. Based on the criterion that the lower boundary of V, V_{min}, is determined by $\Delta\phi_{2I} = \phi_r$ (*cf.* Sec. II-B-1-b), it can be seen that V = 8.6 volts is close to V_{min}. By repeating the computation with other values of V it was found that V_{min} = 7.1 volts. This value agrees with experimental observations.

c. METHODS OF COMPUTATION

The results of computation using the Runge-Kutta and the Adams methods of solution were practically identical with the results obtained by using the simple method of computation. Hence, the results of either method may be trusted to be essentially the true solutions, *i.e.*, the computational errors are negligible.

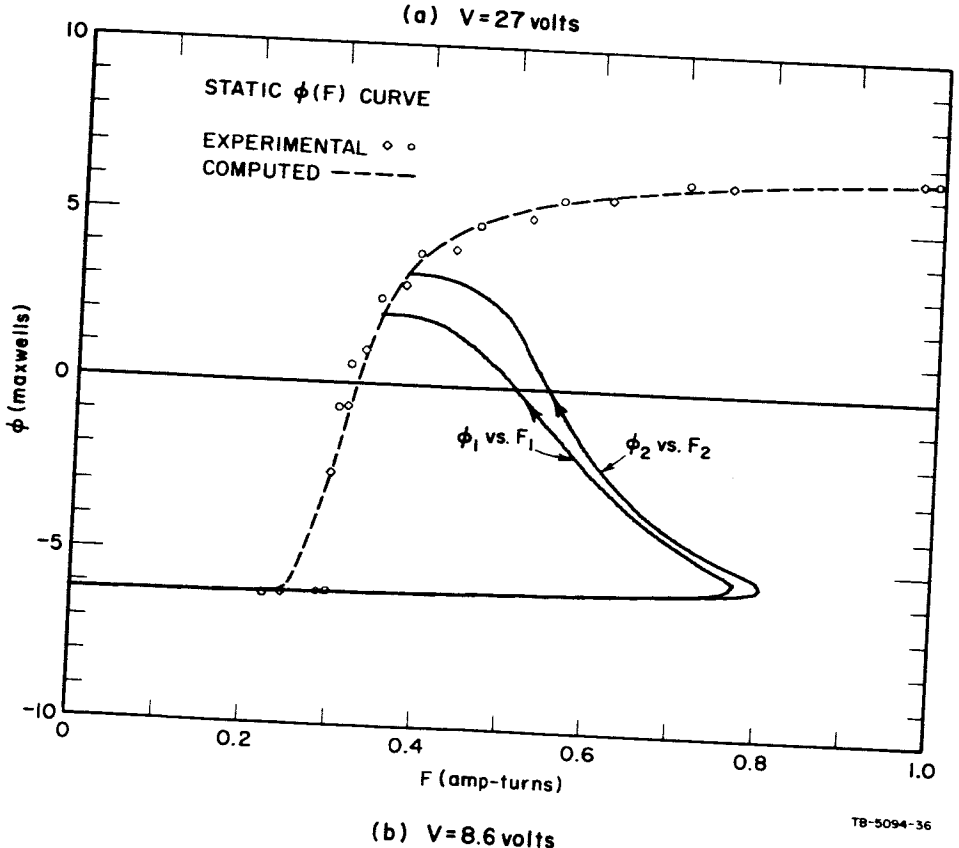
Although library computer PROCEDUREs are available for both the Runge-Kutta and the Adams methods of solution, the application of these PROCEDUREs is not easier (and may even be harder) than writing the whole computer program from scratch and using the simple predictor-corrector method. This is so because most of the computation steps using the simple method, which revolves around the transcendental solution of i_d and i_s, as well as the differential equations, must be provided in the $F(X, Y, DX)$ PROCEDURE. In addition, for the same specified accuracy, the computation time (including compiling time) using the simple method is much shorter (approximately by a factor of 4) than the one using the Runge-Kutta and the Adams methods.

Since the results of the two methods of computation are essentially the same, and since the simple method requires much less computer time for the same accuracy, we conclude that, for problems of this type, the simple predictor-corrector method of solution is preferred. We expect this to be true for even more complicated magnetic circuit problems.



(a) V = 27 volts

TB-5094-35



(b) V = 8.6 volts

TB-5094-36

FIG. 23 VARIATIONS OF ϕ_1 vs. F_1 AND ϕ_2 vs. F_2 DURING MODE I OF A CORE-DIODE-TRANSISTOR BINARY COUNTER

7. Summary

The operation of a core-diode-transistor binary counter is described by dividing a two-count period into four modes of operation; Mode I, during which both cores switch simultaneously relatively fast; and Modes II, III, and IV, during which one core at a time switches relatively slowly. A computational analysis of Mode I is provided for evaluating five time variables (ϕ_1, ϕ_2, i_L, i_s, and i_d) by solving five implicit equations, three of which are differential. The computation involves Newton's method for the transcendental solution of i_s and i_d, in addition to a predictor-corrector method for the solution of a set of differential equations. Two methods of solution are applied, a simple one and the Runge-Kutta and Adams method. The results are practically identical, and the agreement with experimental data is satisfactory. The minimum supply voltage is computed, and its value is in agreement with experimental observation.

C. FLUX DIVISION IN A LOADED SATURABLE CORE

1. Introduction

The problem of flux division in a multipath core was described in Report 2 (pp. 24-48; 83-95). Calculation of flux division in a loaded saturable three-leg core, which is initially in a CLEAR state, was carried out. The calculation was based on the simple switching model $\dot{\phi} = \dot{\phi}_p[1 - (\phi/\phi_s)^2]$, which is valid only if F is large enough to switch ϕ from $\phi = -\phi_r$ to $\phi = +\phi_s$. Despite the simplicity of this model (relative to the more exact models we now employ), the calculation involved a large amount of algebraic manipulations and, except for very high drive current, instantaneous values of F had to be approximated by time-averaged values of F. The agreement with experimental data for low drive-current amplitude was not satisfactory because of the limitations of the model. In addition, it was found later[12] that the experimental data, which had been recorded at a temperature of $29 \pm 2°C$, was very sensitive to temperature, and as a result had to be recorded again at a temperature that was tightly controlled at $29 \pm 0.5°C$.

Our objective now is to improve the computation of flux division described in Report 2 by overcoming most of the above-mentioned drawbacks. Specifically, we shall employ our latest switching models for $\dot{\phi}_{ma}$ and $\dot{\phi}_i$, we shall avoid the struggle against a complex algebra by resorting to a numerical solution of the basic equations, we shall compute instantaneous

(rather than average) values of the time variables, and we shall compare the results with the more reliable experimental data[12] (taken at 29 ± 0.5°C).

2. Experiment

The core geometry, the circuit, and the drive current involved in the flux-division experiment were given in Fig. 10 of Report 2. For the convenience of the reader, this figure is reproduced in Fig. 24. A three-leg saturable ferrite core, whose main leg (Leg m) has twice the flux capacity of each of the other legs (Legs 3 and 4), is initially in a CLEAR state, *i.e.*, $\phi_m = -2\phi_r$ and $\phi_3 = \phi_4 = -\phi_r$. With Leg 4 loaded by a resistance R_L across N_L turns, a SET drive mmf of constant amplitude, NI, is applied to Leg m, and its duration, T, is adjusted so that $\Delta\phi_m = 2\phi_r$.

We wish to compute the variations of $F(t)$, $\dot{\phi}(t)$, and $\phi(t)$ in each leg during the switching time and then compute the flux division ratio, $D = \Delta\phi_3/\Delta\phi_4$, for various magnitudes of drive current, load, and leg-length ratio, l_4/l_3.

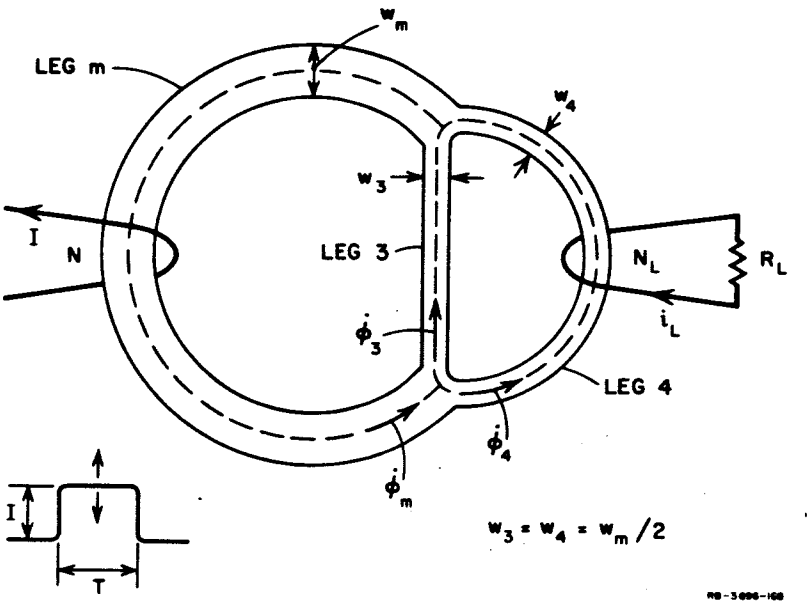


FIG. 24 FLUX DIVISION IN A LOADED SATURABLE THREE-LEG CORE

3. Computational Analysis

a. Basic Equations

Because inelastic flux division is of interest, we shall neglect the elastic component of $\dot{\phi}$, $\dot{\phi}_e$. However, since NI may be relatively low, we have to distinguish between $\dot{\phi}_i$, Eq. (5), and $\dot{\phi}_{ma}$, Eqs. (18), (19), and (21). Only if NI is large enough to switch *all* legs relatively fast can we combine $\dot{\phi}_i$ and $\dot{\phi}_{ma}$ into $\dot{\phi}$ expressed by Eqs. (18) through (20). Designating $\dot{\phi}_{ma}$, $\dot{\phi}_i$, $\dot{\phi}$, ϕ, F, and T_i of Leg m, Leg 3, and Leg 4 by Subscripts m, 3, and 4, respectively, three differential equations are written formally as follows:

$$\dot{\phi}_m = \dot{\phi}_{ma\,m}(F_m,\phi_m) + \dot{\phi}_{i\,m}(F_m,t,T_{i\,m}), \tag{76}$$

$$\dot{\phi}_3 = \dot{\phi}_{ma\,3}(F_3,\phi_3) + \dot{\phi}_{i\,3}(F_3,t,T_{i\,3}), \tag{77}$$

and

$$\dot{\phi}_4 = \dot{\phi}_{ma\,4}(F_4,\phi_4) + \dot{\phi}_{i\,4}(F_4,t,T_{i\,4}) \quad . \tag{78}$$

By inspection of Fig. (24),

$$F_3 = F_4 + (N_L^2/R_L)\dot{\phi}_4, \tag{79}$$

$$F_m = NI - F_3, \tag{80}$$

and

$$\dot{\phi}_m = \dot{\phi}_3 + \dot{\phi}_4 \quad . \tag{81}$$

Equations (76) through (81) contain six dependent variables: ϕ_m, ϕ_3, ϕ_4, F_m, F_3, and F_4. We now wish to solve for these variables numerically.

b. Method of Computation

For a given value of t, the values of $T_{i\,m}$, $T_{i\,3}$, and $T_{i\,4}$ are known, and $\dot{\phi}_4$, Eq. (78), is a function of F_4 and ϕ_4. Therefore, following Eqs. (79) and (80), F_3 and F_m are also functions of F_4 and ϕ_4. When the latter are substituted into Eqs. (76) and (77), we obtain a set of three differential equations of the form

$$\dot{\phi}_m = f_m(F_4, \phi_4, \phi_m), \tag{82}$$

$$\dot{\phi}_3 = f_3(F_4, \phi_4, \phi_3), \tag{83}$$

and

$$\dot{\phi}_4 = f_4(F_4, \phi_4) \quad . \tag{84}$$

Any of the conventional methods of solution of ϕ_m, ϕ_3, and ϕ_4 must be supplemented by the transcendental solution of F_4, which is based on substitution of Eqs. (82) through (84) into Eq. (81). Thus, we are looking for the root F_4 of the equation

$$f = \dot{\phi}_3 + \dot{\phi}_4 - \dot{\phi}_m = 0 \quad . \tag{85}$$

Following Eqs. (79) and (80),

$$-\frac{\partial F_m}{\partial F_4} = \frac{\partial F_3}{\partial F_4} = 1 + \frac{N_L^2}{R_L}\,\dot{\phi}_4' \quad . \tag{86}$$

Differentiation of Eq. (85) with respect to F_4, and substitution of Eq. (86) give

$$\frac{\partial f}{\partial F_4} = f' = (\dot{\phi}_m' + \dot{\phi}_3')\left(1 + \frac{N_L^2}{R_L}\,\dot{\phi}_4'\right) + \dot{\phi}_4' \quad . \tag{87}$$

Following Newton's method, the iterative correction for F_4 is $-f/f'$. However, if the sign of this correction alternates, the correction will be only $-0.5f/f'$.

c. SWITCHING PARAMETERS

The switching parameters of the three legs differ from each other because of differences in leg dimensions (width and length). These parameters are evaluated using the following material parameters (*cf.* pp. 37-41 of Report 2, and Sec. I-B-2-e in this report): B_r, B_s, H_a, H_q, H_n, H_0'', ν, κ, H_0, H_B, ζ_p, ν_i, κ_i, M_i, and H_{0i}.

In analogy to Eqs. (24) and (25) of Report 3 (pp. 17-18), the continuity of $\dot{B}_p$ *vs.* H (*cf.* Fig. 15 in Report 2) at $H = H_B$ requires the following relationships:

$$H_B = \frac{\nu H_0 - H_0''}{\nu - 1} \tag{88}$$

and

$$\zeta_p = \kappa\nu(H_B - H_0'')^{\nu-1} \tag{89}$$

Following Eqs. (88), (89), (91), and (92) of Report 2 (pp. 39-40), Eqs. (26) through (29) of Report 3 (pp. 18-19) and Eqs. (14) through (16) of this report, the switching parameters of a leg are expressed by the following functions of material parameters and leg dimensions [l_i, l_o, $l = (l_i + l_o)/2$, and $A = hw$]: $\phi_r = AB_r$; $\phi_s = AB_s$; $F_0'' = lH_0''$; $F_B = lH_B$; $F_0 = lH_0$; $\rho_p = \zeta_p A/l$; $\lambda = \kappa A/(l^\nu)$; $F_{12} = H_q l_i$; $F_{23} = H_q l_o$; $V_1 = (\phi_s - \phi_r)/[H_a(l_o - l_i)]$; $V_2 = (\phi_s + \phi_r)\, H_q/[H_n(l_o - l_i)]$; $\lambda_i = \kappa_i A/(l^{\nu_i})$; $C_i = M_i l$; and $F_{0i} = H_{0i} l_i$. These expressions are used to compute the switching parameters of Leg m, Leg 3, and Leg 4.

d. ESTIMATION OF τ_s FOR DETERMINATION OF Δt

In most of the magnetic circuit computer programs described so far, we have determined Δt as a given fraction of an estimated switching time, τ_s, *e.g.*, $\Delta t = \tau_s/200$. The value of τ_s was calculated roughly using basically the relation $2\phi_r = \int_0^{\tau_s} \bar{\rho} F_{ex}\, dt$, where $\bar{\rho} \cong 0.6\rho_p$ is the average value of ρ, and $F_{ex} = F - F_0''$ (actually, $F_{ex} = F - F_0$, but in order to obtain a positive value of τ_s if $F_0'' < F < F_0$, F_0 was replaced by F_0''). See, for example, Eqs. (46), (47), and (90) in Report 3(pp. 26 and 63), and Eqs. (72) and (73) in this report. As long as F is not close to F_0'', such an estimation of τ_s, crude as it is, is accepted because the length of Δt is not crucial. However, if F is close to F_0'', the resulting Δt becomes much too short because $\rho_p(F - F_0'') >> \dot{\phi}_p$. In order to get a closer estimate for τ_s, the relation $\dot{\phi}_p = \lambda(F - F_0'')^\nu$ should be used instead of $\dot{\phi}_p = \rho_p(F - F_0'')$. Hence,

$$\tau_s \cong \frac{\phi_r}{0.3\lambda(F - F_0'')^\nu} \tag{90}$$

Equation (90) should be used if $F_0'' \leq F \leq F_B$, but may also be used if F is not much larger than F_B.

Suppose that Leg 4 is so long that $\dot{\phi}_m = \dot{\phi}_3$. In this case, the switching path is composed of Legs m and 3 in series. If NI is not much larger than $F''_{0m} + F''_{03}$, then H_m or H_3 is not much larger than H''_0, and the difference between H_m and H_3 (*cf.* Report 2, Fig. 16, p. 43) may be neglected. Under this condition, the relation $\dot{\phi} = \dot{\phi}_m = \dot{\phi}_3$ leads to an approximate expression for λ of Legs m and 3 in series:

$$\lambda = \frac{1}{\left(\frac{1}{\lambda_m^{1/\nu}} + \frac{1}{\lambda_3^{1/\nu}}\right)^{\nu}} . \tag{91}$$

Substitutions of Eq. (91), $F = F_m + F_3 = NI$, and $\phi_r = \phi_{rm}/2$ into Eq. (90) give

$$\tau_s \cong \frac{\phi_{rm}}{0.6} \left(\frac{\frac{1}{\lambda_m^{1/\nu}} + \frac{1}{\lambda_3^{1/\nu}}}{NI - F''_{0m} - F''_{03}} \right)^{\nu} . \tag{92}$$

e. DRIVE CURRENT

In view of the fact that the drive current has a finite rise time, T_r, we shall replace NI in Eq. (80) by Ni, where

$$Ni = \begin{cases} \frac{NI}{2}\left(1 - \cos \pi \frac{t}{T_r}\right) & \text{if} \quad t < T_r \\ NI & \text{if} \quad t \geq T_r \end{cases} . \tag{93}$$

4. COMPUTER PROGRAM

a. $\dot{\phi}$ PROCEDUREs

The switching parameters of the three legs are different (because of the difference in dimensions). Hence, six separate PROCEDUREs are needed in order to compute $\dot{\phi}_i$ and $\dot{\phi}_{ma}$ of each leg. In order to save program writing, DEFINE declarations[13] are made only once for the steps involved in the $\dot{\phi}_i(F, t, T_i)$ PROCEDURE, Appendix B, and the $\dot{\phi}_{ma}(F, \phi, \phi_d, \dot{\phi}')$ PROCEDURE, Appendix C. Each of these declarations is used in a corresponding $\dot{\phi}$ PROCEDURE of a given leg after its identifiers have been identified

(by using DEFINE declarations) with the corresponding identifiers of the given leg. For example, the $\dot{\phi}_{im}(F_m, t, T_{im})$ PROCEDURE is generated by identifying F_m with F, ϕ_m with ϕ, etc., and by calling the identifier of the DEFINE declaration in which the steps involved in the $\dot{\phi}_i(F, t, T_i)$ PROCEDURE are included [alternatively, we may call the $\dot{\phi}_i(F, t, T_i)$ PROCEDURE itself, but this results in a longer running time of the computer].

There are other ways to achieve the same goal. For example, the switching parameters may be declared as formal input parameters of the $\dot{\phi}$ PROCEDURE (in addition to the existing ones). If the $\dot{\phi}$ PROCEDURE of each leg is called (in the main program) more than twice, say, this alternative is too lengthy. Another alternative is to fill three arrays, one for each leg, with the switching parameters, and to replace each switching parameter in the declaration of the $\dot{\phi}$ PROCEDURE by the corresponding address in the array. The main drawback of this alternative is that it is hard to associate by inspection a given array address with the name of the corresponding switching parameter.

b. MULTIPLE OUTPUT

Three types of output may be needed from the computer program of flux division:

(1) Flux switching in each leg *vs.* time for given values of N_L^2/R_L, l_4/l_3, and NI;

(2) D *vs.* NI, with N_L^2/R_L as a parameter, for a given value of l_4/l_3; and

(3) D *vs.* l_4/l_3, with N_L^2/R_L as a parameter, for a given value of NI.

Which of these output types is to be executed depends on the value 1, 2, or 3 assigned to an identifier *SW* in the program.

c. PROGRAM OUTLINE

The computer program for flux division is given in Appendix I, using the language ALGOL-60. The program outline is as follows.

(1) Declare all identifiers (material parameters, leg dimensions, switching parameters, circuit parameters, variables, and miscellaneous), output lists and formats.

(2) Declare the definitions of $\dot{\phi}_i$ (Appendix B) and $\dot{\phi}_{ma}$ (Appendix C). Declare the following PROCEDUREs: $\dot{\phi}_{im}$, $\dot{\phi}_{maa}$, $\dot{\phi}_{i3}$, $\dot{\phi}_{ma3}$, $\dot{\phi}_{i4}$, $\dot{\phi}_{ma4}$, and tanh (X_1). Define "mm" as 10^{-3} (in order to maintain MKS units while writing leg dimensions in mm).

(3) Set *SW* to 1, 2, *or* 3, depending on the type of output required.

(4) Set the values of the material parameters.

(5) Set and compute the dimensions and switching parameters of Legs *m* and 3. Set and compute w_4, A_4, ϕ_{r4}, and ϕ_{s4}.

(6) If $SW = 3$, print heading for type-3 output.

(7) For each of several values of $S = l_4/l_3$, do the following:

(A) Complete computation of the dimensions and switching parameters of Leg 4.

(B) If $SW = 2$, print heading for type-2 output.

(C) Set the value of the rise time, T_r, of the drive current.

(D) For each of several values of N_L^2/R_L, do the following:

1. For each of several values of *NI*, do the following:

(i) If $SW = 1$, print heading for type-1 output.

(ii) Set the initial conditions ($t = 0$; $\phi_m = -\phi_{rm}$; $\phi_3 = -\phi_{r3}$; $\phi_4 = -\phi_{r4}$; $F_m = F_3 = F_4 = 0$; $\dot{\phi}_m = \dot{\phi}_3 = \dot{\phi}_4 = 0$).

(iii) Compute τ_s, using Eq. (92). Set $\Delta t = \tau_s/200$.

(iv) For each Δt during switching time, do the following

(a) Compute $t = t_{(-1)} + \Delta t$. Compute *Ni*, using Eq. (93).

(b) Lower the Δt index of previously computed ϕ_m, $\dot{\phi}_m$, ϕ_3, $\dot{\phi}_3$, ϕ_4, $\dot{\phi}_4$, and F_4.

(c) Use the relation $y \cong 2y_{(-1)} - y_{(-2)}$ to predict a first approximation for $y = \phi_m$, ϕ_3, ϕ_4, and F_4

(d) Until a specified convergence condition is achieved, compute the following steps in a *loop*; for each leg, use the relation $F_i = F_{0i} \tanh (F/F_{0i})$, Eq. (17), call the corresponding $\dot{\phi}_{ma}$ and $\dot{\phi}_i$ PROCEDUREs, and correct ϕ according to $\phi = \phi_{(-1)} + \Delta t[\dot{\phi} + \dot{\phi}_{(-1)}]/2$, Eq. (31):
F_4(previous F_4 plus δF_4); F_{i4};
$\dot{\phi}_4 = \dot{\phi}_{ma4} + \dot{\phi}_{i4}$; F_3[Eq. (79)]; F_{i3};
$\dot{\phi}_3 = \dot{\phi}_{ma3} + \dot{\phi}_{i3}$; ϕ_3; F_m[Eq. (80)];
F_{im}; $\dot{\phi}_m = \dot{\phi}_{mam} + \dot{\phi}_{im}$; ϕ_m; f[Eq. (85)];
$\dot{\phi}'_m = \dot{\phi}'_{mam} + \dot{\phi}'_{im}$; $\dot{\phi}'_3 = \dot{\phi}'_{ma3} + \dot{\phi}'_{i3}$;
$\dot{\phi}'_4 = \dot{\phi}'_{ma4} + \dot{\phi}'_{i4}$; f'[Eq. (87)];
$\delta F_4 = -f/f'$ (if δF_4 changes sign, $\delta F_4 = -0.5\ f/f'$). Repeat the above steps if $|f| > 0.0001\ \phi_{rm}/\tau_s$, but no more than 19 times. Count the number of times this convergence condition fails to be satisfied.

(v) If $SW = 1$, print type-1 output (t, $\dot{\phi}_m$, ϕ_m, F_m, $\dot{\phi}_3$, ϕ_3, F_3, $\dot{\phi}_4$, ϕ_4, F_4, Ni, number of iterations) once, say, every fifth Δt.

(vi) If $\phi_m > -0.99\ \phi_{rm}$, terminate switching when either $\phi_m \geq 0$ or $\dot{\phi}_m < 0.0001$ volt/turn.

(vii) Compute the net flux changes during the switching time: $\Delta\phi_3 = \phi_3 + \phi_{r3}$ and $\Delta\phi_4 = \phi_4 + \phi_{r4}$.

(viii) Compute flux-division ratio, $D = \Delta\phi_3/\Delta\phi_4$.

(ix) If $SW = 2$, store the value of D in a one-dimensional array (at a location corresponding to the value of NI). If $SW = 3$, store the value of D in a

two-dimensional array (at a location corresponding to the values of N_L^2/R_L and l_4/l_3).

2. If SW = 2, print type-2 output (N_L^2/R_L, D *vs.* NI, and the cumulative number of convergence failures). If SW = 3, store the values of N_L^2/R_L and the cumulative number of convergence failures in the above two-dimensional array.

(E) If SW = 3, print type-3 output (for each value of N_L^2/R_L, D *vs.* l_4/l_3 and the cumulative number of convergence failures).

5. Experimental Verification

a. Core

The saturable core used in the flux-division experiment was referred to as Core S in Report 2 (Fig. 22, p 62). The core was cut ultrasonically from a 1/2-inch-diameter ferrite disk whose material is the same as that of Core E-6: Telemeter Magnetics, T-5, of nominal composition $[Mg_{0.32}Zn_{0.1}Mn_{0.58}]^{++}[Mn_{0.26}Fe_{0.74}]_2^{+++}0_4$. The material parameters of Core S at T = 29°C are given as follows: B_r = 0.23 wb/m^2; B_s = 0.2484 wb/m^2; H_a = 310 amp-turns/m; H_q = 35.0 amp-turns/m; H_n = 30.0 amp-turns/m; H_0'' = 40.0 amp-turns/m; ν = 1.30; κ = 3400 ohms/turn$^{2.3}$ amp$^{0.3}$m$^{0.7}$; H_0 = 61.0 amp-turns/m; H_B = 131.0 amp-turns/m [Eq. (88)]; ζ_p = 17,105 ohms/turn^{2}m [Eq. (89)]; ν_i = 1.3; κ_i = 592 ohms/turn$^{2.3}$ amp$^{0.3}$m$^{0.7}$; M_i = 10.7 amp-turns-μsec/m; and H_{0i} = 24.8 amp-turns/m. The dimensions and computed switching parameters of each leg are given in Table I.

b. Circuit Drive

Four values of N_L^2/R_L (in turns^2/ohm) were tested in the experiment (Fig. 24): 0 (no load), 1.000 (N_L = 1; R_L = 1.00 ohm), 3.962 (N_L = 2; R_L = 1.01 ohm), and 9.524 (N_L = 2; R_L = 0.42 ohm). The temperature was kept at 29 ± 0.5°C. The drive-current rise time was near 0.08 μsec and the mmf amplitude NI was varied between 1.1 amp-turn and 4.0 amp-turns.

Table I

LEG DIMENSIONS AND SWITCHING PARAMETERS OF CORE S

LEG	LEG m	LEG 3	LEG 4
Dimensions			
h (mm)	1.31	1.31	1.31
l_i (mm)	14.363	4.310	7.887
l_o (mm)	19.151	5.108	9.348
l (mm)	16.757	4.709	8.617
w (mm)	1.016	0.508	0.508
A (mm^2)	1.33	0.665	0.665
Switching parameters			
ϕ_r (maxwells)	30.612	15.306	15.306
ϕ_s (maxwells)	33.061	16.531	16.531
λ ($ohms/turn^{2.3}amp^{1.3}$)	0.921	2.398	1.093
F_0'' (amp-turn)	0.670	0.188	0.345
ρ_p ($ohms/turn^2$)	1.359	2.417	1.321
F_0 (amp-turn)	1.022	0.287	0.526
F_B (amp-turns)	2.195	0.617	1.129
λ_i ($ohm/turn^{2.3}amp^{1.3}$)	0.160	0.417	0.190
C_i (amp-turn-μsec)	0.179	0.050	0.092
F_{0_i} (amp-turn)	0.356	0.107	0.196

c. EXPERIMENTAL AND COMPUTED $\dot{\phi}(t)$ WAVEFORMS

A comparison is made in Fig. 25 between experimental and computed waveforms of $\dot{\phi}_m(t)$, $\dot{\phi}_3(t)$, and $\dot{\phi}_4(t)$ for $N_L^2/R_L = 0$ and $NI = 2.0$ amp-turns. The computation was performed on a Burroughs B-5500 computer. The computed waveforms were drawn manually, using type-1 output results. The ringing in the $\dot{\phi}(t)$ oscillograms was caused by the ringing in the drive MMF $Ni(t)$, which is also shown in Fig. 25.

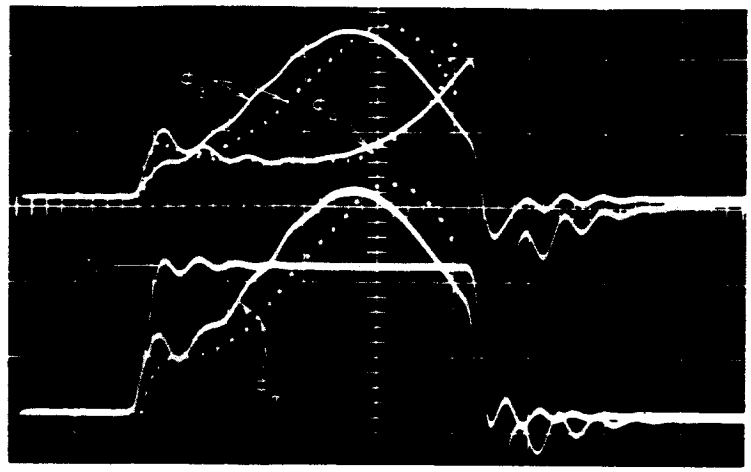


FIG. 25 EXPERIMENTAL (SOLID LINE) AND COMPUTED (DOTTED LINE) Ni(t) AND $\dot{\phi}(t)$ WAVEFORMS OF CORE S ($l_4/l_3 = 1.83$), DRIVEN UNLOADED BY MMF NI OF 2.0 AMP-TURNS IN A FLUX-DIVISION EXPERIMENT
Time scale = 0.2 μsec/major div.; Ni scale = 1.0 amp-turn/major div.; $\dot{\phi}_m$ scale = 0.2 volt/turn/major div.; $\dot{\phi}_3$ scale = 0.25 volt/turn/major div.; $\dot{\phi}_4$ scale = 0.05 volt/turn/major div.

d. MEASURED AND COMPUTED *D vs. NI* WITH N_L^2/R_L AS A PARAMETER

A comparison is made in Fig. 26 between measured and computed *D vs. NI* for N_L^2/R_L = 0, 1.000, 3.962, and 9.524 turns^2/ohm. The computed curves were drawn manually, using type-2 output results.

e. COMPUTED *D vs.* l_4/l_3 WITH N_L^2/R_L AS A PARAMETER

Computed curves of *D vs.* l_4/l_3 are shown in Fig. 27 for four values of N_L^2/R_L (0,5,10, and 25 turn^2/ohm) and two values of *NI* (10 and 100 amp-turns). The core is assumed to be identical with Core S, except for the length of Leg 4. The plots were drawn manually, using type-3 output results.

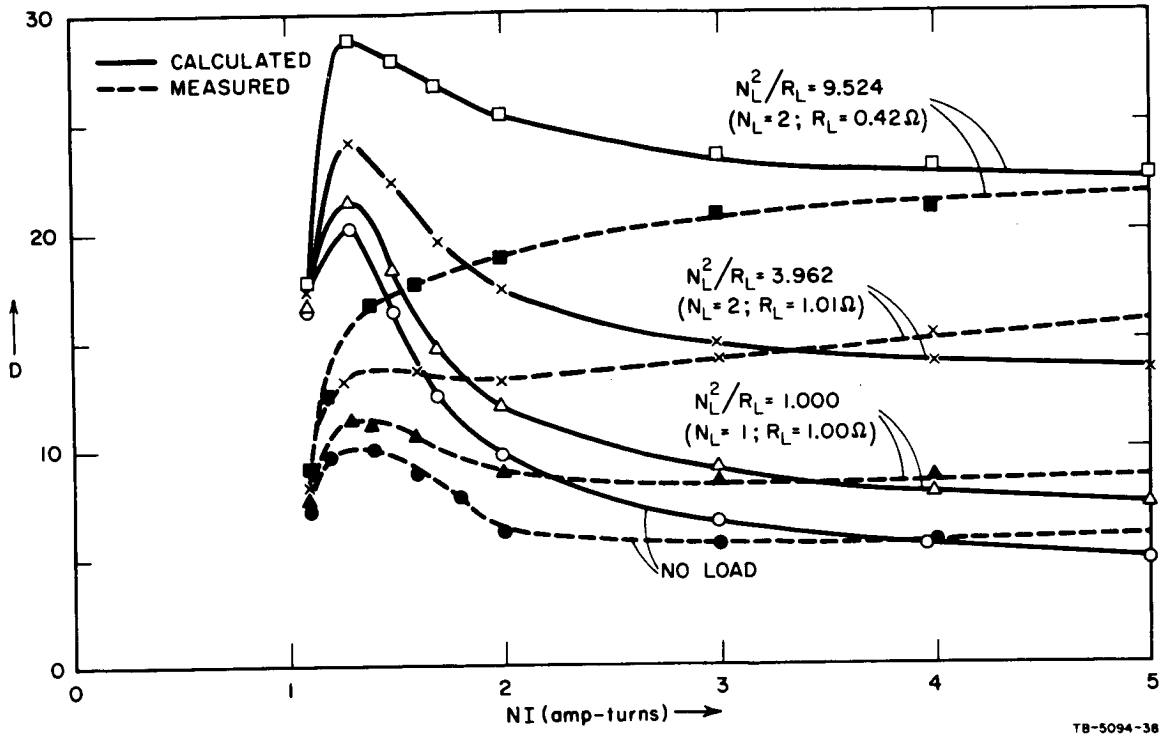


FIG. 26 MEASURED AND COMPUTED FLUX-DIVISION RATIO vs. MMF WITH LOAD AS A PARAMETER FOR CORE S ($l_4/l_3 = 1.83$)

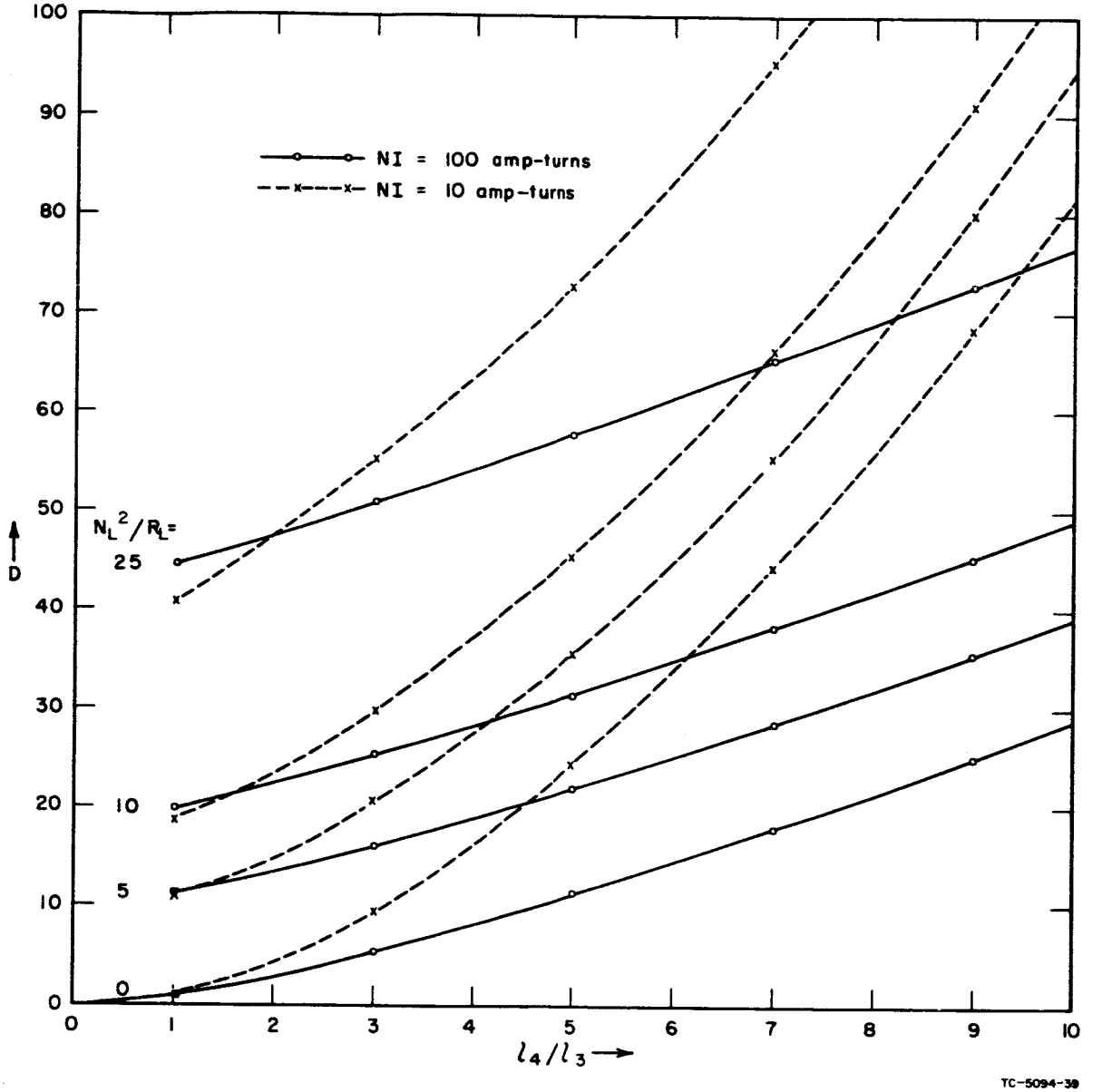


FIG. 27 COMPUTED PLOTS OF FLUX-DIVISION RATIO vs. LEG-LENGTH RATIO WITH LOAD AS A PARAMETER AND NI OF 10 AND 100 AMP-TURNS FOR A CORE IDENTICAL WITH CORE S EXCEPT FOR THE LENGTH OF LEG 4

6. Discussion

a. $\dot{\phi}(t)$ waveforms

The agreement between experimental and computed $\dot{\phi}_m(t)$, $\dot{\phi}_3(t)$, and $\dot{\phi}_4(t)$ in Fig. 25 is very satisfactory. The slight disagreement may be attributed to the difference between the actual waveform of the drive current and the assumed function for $i(t)$. The experimental initial $\dot{\phi}$ spikes are higher than the computed ones. This may be attributed to the omission of the $\dot{\phi}_\epsilon$ component from the computation.

The frequency of ringing in the $Ni(t)$ is the same as in the $\dot{\phi}(t)$ of each leg. We thus conclude that the $\dot{\phi}(t)$ ringing is not pickup but actual variations in $\dot{\phi}(t)$. Elimination of the drive-current ringing would probably result in elimination of the $\dot{\phi}(t)$ ringing.

b. D *vs.* NI with N_L^2/R_L as a parameter

The agreement between experimental and computed D *vs.* NI for the different loads in Fig. 26 is much better than the agreement in Report 2 (Fig. 36, p. 88) in the entire range of NI, and is better than the agreement in Ref. 12 for low values of NI.* Let us distinguish between two regions of NI: NI between 1.1 and 2.1 amp-turns, and NI above 2.1 amp-turns.

In the region of high NI, the agreement is better than in Report 2, but about the same as in Ref. 12. This may be attributed to two factors. First, the validity of the experimental data in Report 2 is in doubt because the temperature was not regulated tightly enough (29 ± 2°C compared with 29 ± 0.5°C in Ref. 12). Second, the switching model used in Report 2 and Ref. 12, $\dot{\phi} = \dot{\phi}_p[1 - (\phi/\phi_s)^2]$, is valid only for high F values. This can be seen by letting $\phi_d = \phi_s$ in $\eta(\phi)$ of Eq. (20) (which may be used to compute $\dot{\phi}_i + \dot{\phi}_{ma}$ if F is high enough).

Consider now the region of low NI. Although there is room for further improvement in the agreement between computed and experimental results in Fig. 26, it should be noted that both reach peak at NI in the neighborhood of 1.3 amp-turns. In contrast, in Report 2 and Ref. 12,

* For the same values of N_L^2/R_L, the computed D *vs.* NI curves in Report 2 and Ref. 12 are identical, but the experimental curves are different due to difference in temperature tolerances.

computed D approaches infinity, and the actual behavior could be explained only qualitatively. Infinite D resulted from the use of the model $\dot{\phi} = \dot{\phi}_p [1 - (\phi/\phi_s)^2]$, which is invalid because the flux switching is incomplete. When this model was later replaced by the model $\dot{\phi} = \dot{\phi}_p \{1 - [(2\phi + \phi_s - \phi_d)/(\phi_s + \phi_d)]^2\}$, Eqs. (18) through (20), peaking D *vs.* NI curves, similar to the ones in Fig. 26 but of higher values, were obtained. The separation between $\dot{\phi}_{ac}$ and $\dot{\phi}_i$ done here has resulted in a still better agreement (lower peaks). A still further improvement would probably result if the slow-switching component of $\dot{\phi}_i$ were added, because $\dot{\phi}_i$ has an effect of increasing $\Delta\phi_4$ and thus decreasing D.

It should be emphasized that examining the validity of a switching model by comparing experimental and computed D is a very severe test if D is high (in which case $\Delta\phi_4$ is small). A slight error in $\Delta\phi_4$, which is negligible in the absolute sense, will have a large effect on D. [On the other hand, in general, comparing $\Delta\phi$ is a less severe test than comparing $\dot{\phi}(t)$].

c. D *vs.* l_4/l_3 WITH N_L^2/R_L AND NI AS PARAMETERS

Let us examine the computed plots of D *vs.* l_4/l_3 in Fig. 27. For each load case, the plots corresponding to NI of 10 amp-turns and of 100 amp-turns intersect. The values of l_4/l_3 corresponding to the intersection points increase from 1 to 2 as N_L^2/R_L increases from zero to 25. This phenomenon is quite complex and may be explained qualitatively by examining two cases of l_4/l_3 values, a large one (*e.g.*, 5) and a small one (*e.g.*, 1). Since both Leg 3 and Leg 4 switch during the same switching time T (*cf.* Fig. 24), the higher the ratio $\dot{\phi}_{p3}/\dot{\phi}_{p4}$ (or $\dot{B}_{p3}/\dot{B}_{p4}$, since $A_3 = A_4$) is, the larger is $\Delta\phi_3/\Delta\phi_4 = D$.

Consider the case of large l_4/l_3 first. Because $l_4 >> l_3$ and because of the load on Leg 4, $\bar{H}_3 >> \bar{H}_4$ for either NI = 10 amp-turns or NI = 100 amp-turns, where $\bar{H}$ designates a time average value of $H(t)$. As NI increases from 10 to 100 amp-turns, the ratio $\dot{B}_{p3}/\dot{B}_{p4}$ decreases *because of the nonlinearity in the* $\dot{B}_p$ *vs.* H *curve* ($\bar{H}$ is assumed smaller than H_B; see Fig. 15, p. 38, in Report 2). Hence, as NI increases, D decreases, which agrees with the results in Fig. 27. Note, however, that the decrease in $\dot{B}_{p3}/\dot{B}_{p4}$ is *counteracted by the load current* which, in comparison with the no-load case, lowers $\bar{H}_4$. This explains why, for

a given value of l_4/l_3 in Fig. 27, the percentage decrease in D (as NI increases from 10 amp-turns to 100 amp-turns) is smaller the higher the value of N_L^2/R_L is.

In the case of $l_4/l_3 = 1$, $\bar{H}_3 > \bar{H}_4$ only because of the load on Leg 4. For the same load, therefore, $\bar{H}_4$ is higher than in the case of a large l_4/l_3, and may exceed H_B if NI is high enough. Hence, as NI increases from 10 amp-turns to 100 amp-turns, the effect of the nonlinearity of $\dot{B}_p$ *vs.* H on decreasing D is overcome by the effect of the loading current on increasing D.

We see that, as NI increases, there are two opposing effects on D: an increase in D due to the load current, and a decrease in D due to the nonlinearity of $\dot{B}_p(H)$. If $l_4/l_3 = 1$, the effect of the load current predominates and D increases with NI. As l_4/l_3 is increased, $\bar{H}_4$ becomes smaller and the nonlinearity of $\dot{B}_p(H)$ becomes more effective, until beyond a certain value of l_4/l_3, the latter overcomes the effect of the load current and D decreases with NI.

Extrapolation of the intersection points of the curves corresponding to 10 and 100 amp-turn indicates that for Core S ($l_4/l_3 = 1.83$), D decreases as NI increases if $N_L^2/R_L < 20$. This is in agreement with the plot of D *vs.* NI in Fig. 26 and the case of $N_L^2/R_L = 40$ in Report 2, Fig. 36(b) (p. 88), where D increases as NI increases above 3 amp-turns. Note, however, that the experimental data in Fig. 26 show that D increases with NI for a value of N_L^2/R_L lower than 20.

7. Summary

An analysis is presented for computing flux-division ratio, D, in a saturable core having three legs: Leg m, Leg 3, and Leg 4. Leg m is driven by mmf NI, and Leg 4 is loaded by a resistance R_L across N_L turns. The analysis is based on six equations: two mmf equations, one junction equation ($\Sigma\dot{\phi} = 0$), and three differential equations ($\dot{\phi} = \dot{\phi}_{ma} + \dot{\phi}_i$ for each leg). A numerical solution for $\phi(t)$ and $F(t)$ is obtained by incorporating a transcendental solution for F_4 into a simple predictor-corrector method. A computer program is provided for computing three types of output: time variables, D *vs.* NI and N_L^2/R_L for given l_4/l_3, and D *vs.* l_4/l_3 and N_L^2/R_L for given NI. Computed $\phi(t)$ waveforms for $NI = 2.0$ amp-turns and D *vs.* NI for $N_L^2/R_L = 0$, 1.000, 3.962, and 9.524 turns^2/ohm are

compared with experimental data from an ultrasonically cut ferrite core (Core S, Report 2). The agreement between computed and experimental results is considerably better than the agreement in Report 2.

D. CONCLUSIONS

The complex nonlinearity of exact flux-switching functions makes it impossible to complete the analysis of magnetic circuits algebraically. Even if these functions are simplified, the amount of mathematical work involved is very often a burden. These two factors dictate the use of numerical methods in order to analyze (and design) magnetic circuits.

Typically, the problem is to solve a set of simultaneous first-order differential equations that include variables that must be solved for transcendentally. For the transcendental solution of these variables, Newton's method (with a slight modification) has been found to be very effective. The set of differential equations may be solved by a number of well-known methods. In the case of the binary counter (which is not a simple circuit), the results obtained by using a simple predictor-corrector method are essentially the same as the results obtained by a more complex and more exact method (Runge-Kutta and Adams); on the other hand, the computation time using the simple method was much shorter. The simple method has also yielded results that agree quite well with experimental data for other types of magnetic circuits (unloaded core, loaded core, core-diode shift register, and a loaded saturable three-leg core). We thus find the simple predictor-corrector method to be accurate enough for the computational analysis of magnetic circuits similar to the ones above and to be relatively inexpensive.

The agreements between computed and experimental waveforms of current and voltage in the circuits analyzed so far are encouraging. Future work along this line includes the following areas:

(1) Computational analyses (including computer programs and experimental verification) of additional, more complicated magnetic circuits.

(2) Computer programs for determination of core parameters from experimental data.

(3) Computer programs for *design* of magnetic circuits.

(4) Application of on-line computers in analysis and design of magnetic circuits.

III VARIATIONS OF FLUX-SWITCHING PARAMETERS

A. RAMP-F SWITCHING

1. INTRODUCTION

It was noted in Report 3 (pp. 35 and 129) that the peak switching voltage of Core E-6 with a ramp-F drive was considerably lower than that computed from the parabolic model using step-F parameters. An investigation has been carried out to determine the extent of this effect as a function of the slope, k, of the ramp F. This effect is probably closely related to the effects of partial setting discussed in Report 3, but the nature of the relationship is not yet understood.

The $\dot{\phi}(t)$ comparison in Report 3 [Fig. 11(a)] showed that the shape of the $\dot{\phi}(t)$ waveform was accurately given by the model for a given slope of F provided that several parameters were appropriately adjusted. The problem is thus primarily one of determining the values of the switching parameters as a function of k. One additional experimental check of the shape of the $\dot{\phi}(t)$ waveform will be given in Sec. III-A-4.

There are two motives for pursuing this study: (1) to determine if the parabolic model can be used for practical applications where a core is driven by an $F(t)$ which is approximately ramp-shaped, and (2) to learn more about the general effects of nonconstant $F(t)$ switching so that, eventually, a new model can be developed which will work for any $F(t)$.

The investigation was carried out by measuring $\dot{\phi}_p$ and t_p for ramp-F switching for a number of values of k (the slope of the ramp). These data were then compared to computed curves to determine if a new, different, set of values of the switching parameters could be used in the model for all values of k.

The experimental $\dot{\phi}_p$ and t_p data were plotted *vs.* k and compared to curves computed from the parabolic model. Curves were computed in two different ways: (1) using a simple version of the parabolic model and solving for $\dot{\phi}_p$ and t_p algebraically, and (2) using the ϕ_d parabolic model and the digital computer. The algebraic method proved useful as an intuitive aid to understanding the results and also provided an accurate

method for determining ramp-F corrected parameter values without resorting to trial and error. These two methods of computing $\dot{\phi}_p$ and t_p will be discussed in Parts A-2 and A-3, respectively. Next, the experimental results of Core E-6 will be briefly discussed. The results of introducing new parameter values into the computations will be discussed in Part A-5. Finally, conclusions will be made in Part A-6. Experimental information for two other cores will be given in Sec. III-B-4 where the effects of temperature are treated.

2. Algebraic Calculation of $\dot{\phi}_p$ and t_p

The values of $\dot{\phi}_p$ and t_p for ramp-F switching cannot be algebraically calculated using the ϕ_d parabolic model. However, they can be algebraically calculated with the following version of the parabolic model:

$$\dot{\phi} = \lambda(F - F_0'')^{\nu}\left(1 - \frac{\phi^2}{\phi_s^2}\right) \quad . \tag{94}$$

Although this model is less accurate than the ϕ_d parabolic model, it is sufficiently accurate to aid intuitively in understanding the phenomenological aspects of ramp-F switching.

It will be assumed for this calculation that the entire $\dot{\phi}_p(F)$ curve can be adequately described with one set of values for λ, F_0'' and ν. This simplifies the calculation considerably and is quite accurate for some cores (*e.g.*, Core E-6) in the range of F over which the experiments were performed. The computer calculation of $\dot{\phi}_p$ and t_p is easily capable of separating the $\dot{\phi}_p(F)$ curve into two regions when it is necessary to do so. The object now is to solve for $\dot{\phi}_p$ and t_p as a function of k (the slope of the applied ramp-F), λ, F_0'' and ν.

The $\dot{\phi}(t)$ function can be obtained from Eq. (94) by separation of variables. Defining t_0 as the time when switching begins, *i.e.*, $\phi = -\phi_r$ at $t = t_0$, and solving for $\dot{\phi}(t)$ results in

$$\dot{\phi} = \lambda(F - F_0'')^{\nu}\,\mathrm{sech}^2\left\{\frac{\lambda}{\phi_s}\int_{t_0}^{t}(F - F_0'')^{\nu}dt' - \tanh^{-1}\frac{\phi_r}{\phi_s}\right\} \quad . \tag{95}$$

Substituting the ramp function

$$F = kt \tag{96}$$

into the integral of Eq. (95) and using the relation $t_0 = F_0''/k$ results in

$$\int_{t_0}^{t} (F - F_0'')^{\nu} dt' = \frac{(kt - F_0'')^{\nu+1}}{k(\nu + 1)} \quad . \tag{97}$$

Using this integral in Eq. (95) gives us the desired $\dot{\phi}(t)$ function,

$$\dot{\phi} = \lambda(F - F_0'')^{\nu} \operatorname{sech}^2 \left\{ \frac{\lambda(kt - F_0'')^{\nu+1}}{\phi_s k(\nu + 1)} - \tanh^{-1} \frac{\phi_r}{\phi_s} \right\} \quad , \tag{98}$$

for ramp-F switching. The solution for t_p is then obtained in the usual way by setting $d\dot{\phi}/dt = 0$. This gives the following transcendental function:

$$(kt_p - F_0'')^{\nu+1} \tanh \left\{ \frac{\lambda(kt_p - F_0'')^{\nu+1}}{\phi_s k(\nu + 1)} - \tanh^{-1} \frac{\phi_r}{\phi_s} \right\} = \frac{\nu k \phi_s}{2\lambda} \quad . \tag{99}$$

It turns out that the argument of the tanh function is usually very small (*e.g.*, < 0.2) so that the approximation

$$\tanh x \simeq x \tag{100}$$

can be used to simplify Eq. (99). This results in a quadratic equation which can be solved for kt_p, thus giving

$$kt_p = \left\{ \frac{k\phi_s(\nu + 1)}{2\lambda} \left[C + \sqrt{C^2 + \frac{2\nu}{\nu + 1}} \right] \right\}^{1/(\nu+1)} + F_0'' \tag{101}$$

where

$$C = \tanh^{-1} \frac{\phi_r}{\phi_s} \quad .$$

The function for $\dot{\phi}_p$ can be obtained by letting $t = t_p$ and $\dot{\phi} = \dot{\phi}_p$ in Eq. (98), and using kt_p from Eq. (101). We thereby obtain

$$\dot{\phi}_p = \lambda \left\{ \frac{k\phi_s(\nu+1)}{2\lambda} \left[C + \sqrt{C^2 + \frac{2\nu}{\nu+1}} \right] \right\}^{\nu/(\nu+1)} \operatorname{sech}^2 \left\{ \frac{1}{2} \left[\sqrt{C^2 + \frac{2\nu}{\nu+1}} - C \right] \right\} . \tag{102}$$

Note that F_0'' does not appear in Eq. (102). In Eq. (101) it appears only as an added term. Thus, two cores which have identical parameters and dimensions except for a difference in F_0'' will both give the same values for $\dot{\phi}_p$. However, the peak of the large F_0'' core will occur later by an amount $1/k$ $(F''_{0\,large} - F''_{0\,small})$. At low values of k this is not precisely true because of the inaccuracies of this model. The F_0'' term in Eq. (101), if divided by k, is equal to the time required for F to reach F_0'', at which time the core begins switching. The time from the onset of switching until the peaking of $\dot{\phi}$, is independent of F_0'' for this model.

The three parameters of primary interest in Eqs. (101) and (102) are k, λ, and F_0''. If everything else is combined into the quantities A and B, these equations can be written as

$$kt_p - F_0'' = A\left(\frac{k}{\lambda}\right)^{1/(\nu+1)} \tag{103}$$

and

$$\dot{\phi}_p = B\lambda\left(\frac{k}{\lambda}\right)^{\nu/(\nu+1)} . \tag{104}$$

Equations (103) and (104) are plotted as a dashed line in Figs. 28 and 29. These curves will be compared with computer calculated curves in Part A-3. The curve is plotted as kt_p *vs.* k rather than t_p *vs.* k to remove the $1/k$ dependence so that the remaining effects can be more easily judged.

If $(kt_p - F_0'')$ of Eq. (103) is plotted *vs.* k on log-log paper, a straight line results. The slope of this line is equal to $1/(\nu+1)$. Vertical displacement of the line corresponds to changes in λ and the coefficient A which contains ν, ϕ_s, and $\tanh^{-1}\phi_r/\phi_s$. Likewise, Eq. (104) gives a straight line on log-log paper. In this case, the slope is equal to $\nu/(\nu+1)$, and vertical displacement of the line corresponds to changes in λ and the coefficient B which also contains ν, ϕ_s and $\tanh^{-1}\phi_r/\phi_s$.

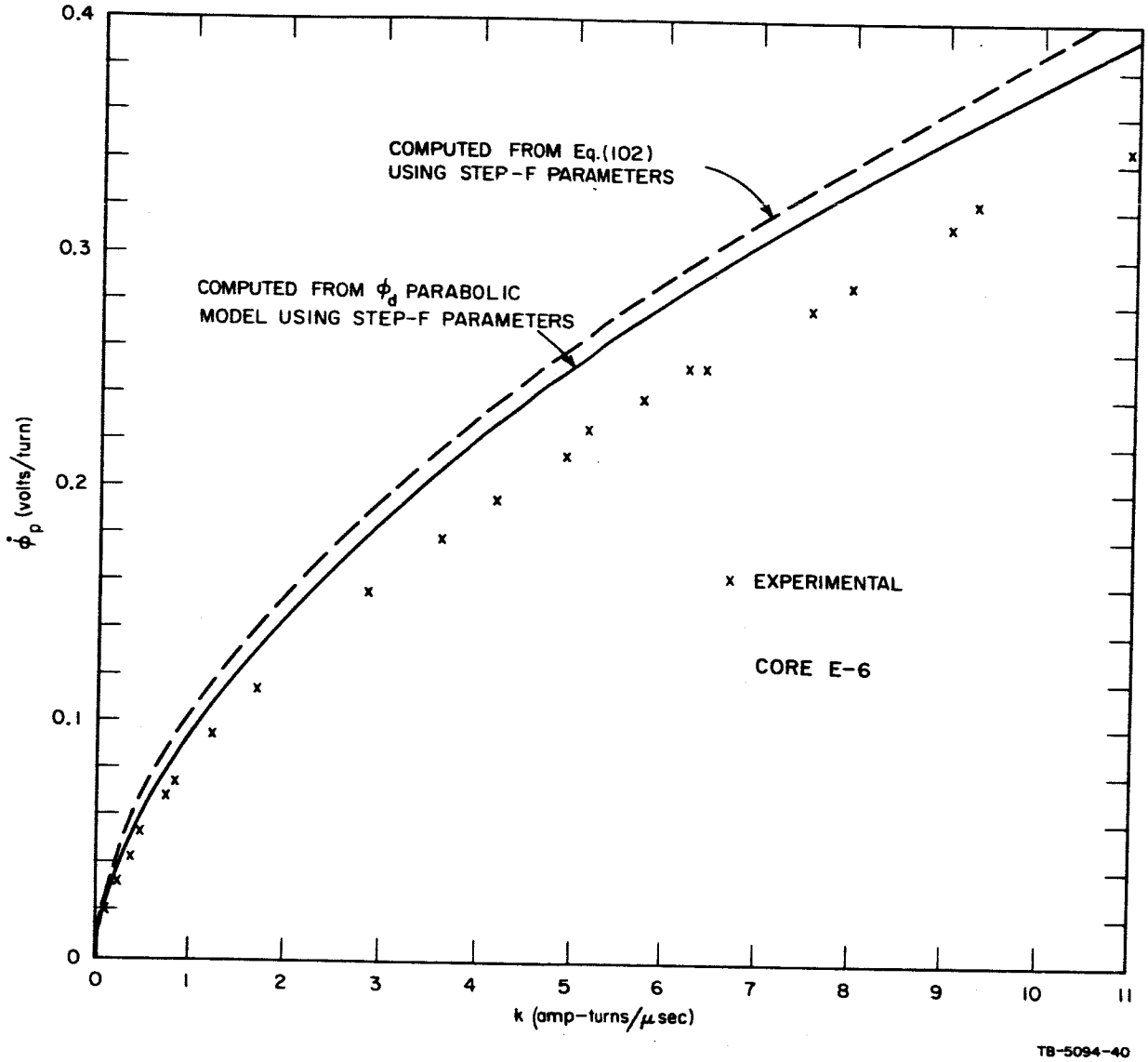


FIG. 28 $\dot{\phi}_p(k)$ FOR RAMP F(t)

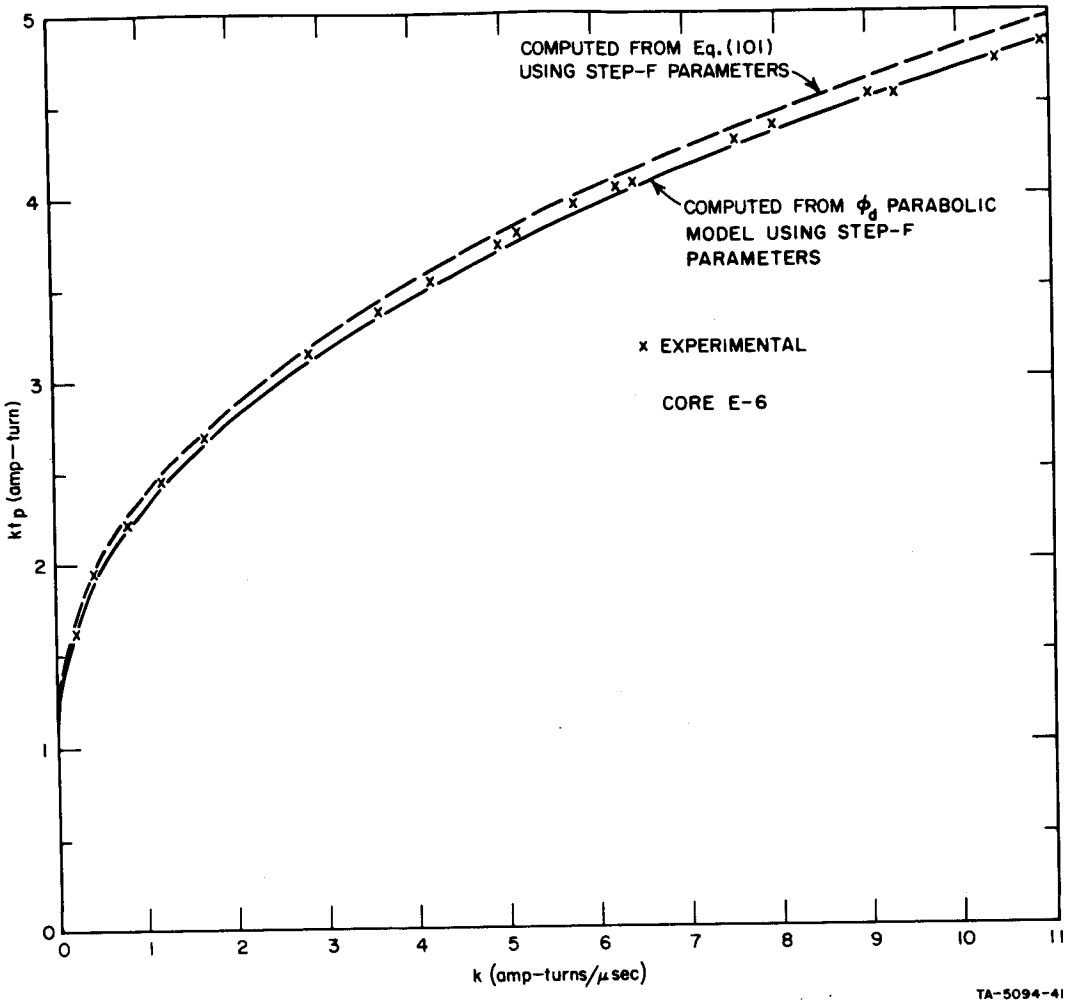


FIG. 29 $kt_p(k)$ FOR RAMP F(t)

The experimental data can be conveniently analyzed on log-log scales because nearly straight lines are obtained. This makes for easier interpolation and simplifies the comparisons with results computed from a model. Equations (103) and (104) will be used in Part A-5 to calculate new values of λ and F''_0 (*e.g.*, λ_r and F''_{0r}) which are applicable specifically for ramp-*F* switching.

3. Numerical Computation of $\dot{\phi}_p$ and t_p

The ϕ_d parabolic model was used with the digital computer to compute more accurate $\dot{\phi}_p(k)$ and $kt_p(k)$ curves than those obtained via the algebraic calculations just discussed. The $\dot{\phi}(t)$ was computed in the vicinity of the maximum $\dot{\phi}$ (*e.g.*, $-0.3\phi_r < \phi < +0.3\phi_r$) for each of a number of k values. The values of $\dot{\phi}_p$ and kt_p were then determined from the $\dot{\phi}(t)$ data and values plotted *vs.* k. The computer program, which did not include

any of the initial spike models, is discussed in some detail on pp. 16-21 of Report 3. The equation of this model which corresponds to Eq. (94) is

$$\dot{\phi} = \lambda(F - F_0'')^{\nu} \left\{ 1 - \left(\frac{2\phi + \phi_s - \phi_d}{\phi_s + \phi_d} \right)^2 \right\} \quad . \tag{105}$$

This model requires a function (see p. 19 of Report 3) for $\phi_d(F)$. Equations for this purpose contain parameters (*e.g.*, ϕ_r, ϕ_s, H_q, H_n, and H_a) that must be determined before this model can be used. These parameters are determined from the static $\phi(F)$ curve.

Computation of $\dot{\phi}_p(k)$ and $kt_p(k)$ was made for Core E-6 using the same step-F parameters as were used in the algebraic calculation above; only one region in $\dot{\phi}_p(F)$ was used. The results are plotted as a solid curve in the linear graphs of Figs. 28 and 29 and also on the log-log graphs of Figs. 30 and 31. Only one region in $\dot{\phi}_p(F)$ was also used for Cores I-4

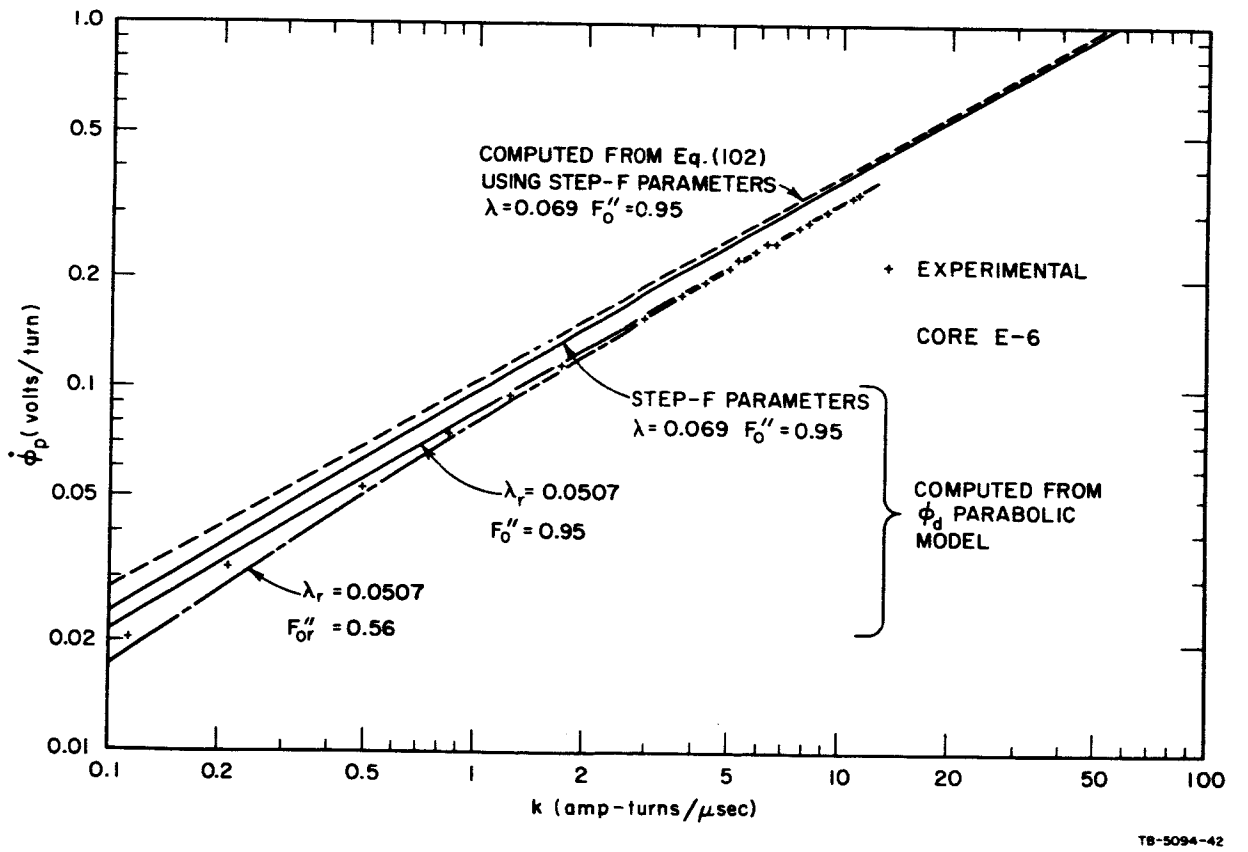


FIG. 30 $\dot{\phi}_p(k)$ FOR RAMP F(t) USING λ_r AND F''_{0r}

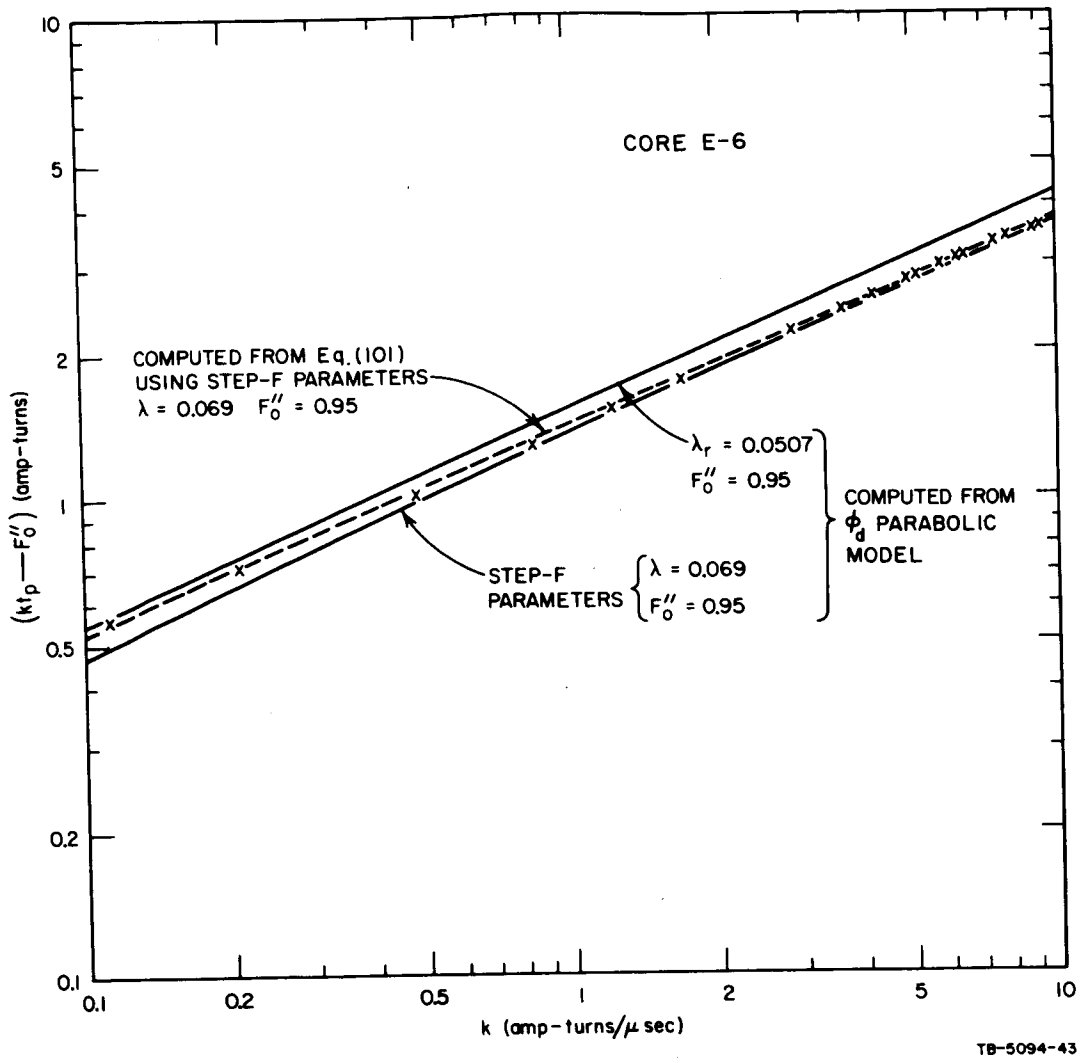


FIG. 31 $(kt_p - F_0'')$ vs. k FOR RAMP F USING λ_r AND F_0''

and K-1 to be discussed in Part B-4 because the experimental $\dot{\phi}_p(F)$ data for step-F drive could be adequately described with only one region. Two regions can easily be handled by the present computer program if necessary.

A comparison of the $\dot{\phi}_p(k)$ curve in Fig. 30 calculated from Eq. (102) and the curve obtained using the computer reveals several interesting points. The curve from the computer is curved downward for low values of k, whereas the algebraic calculation results in a straight line. This curvature is the result of ϕ_d being a function of F in the ϕ_d parabolic model. The computer calculated curves were consistently lower for all

values of k, although as k increased, the percentage difference decreased. For Core E-6 at k = 10, the difference is 4.3 percent; at k = 100, it is 1.2 percent.

4. Experiment

The equipment used for the ramp-F experiments was essentially the same as that described in Sec. I-D-1. The transistor pulser, which consisted of four current drivers in parallel, was modified to increase the maximum rise time to about 50 μsec. A capacitor decade was inserted in the internal pulse-shaping circuit of each of these four drivers. This also increased the fall time, but that is of little consequence if switching is completed during the rise time.

The pulse sequence consisted of a positive SET pulse followed by a negative CLEAR pulse, a positive CLEAR pulse, and a negative CLEAR pulse (see p. 83 of Report 3 for details of this method of clearing). The negative CLEAR pulses were greater than 7 amp-turns for Cores E-6 and K-1, and 15 amp-turns for Core I-4. The constant portion of the SET pulse was kept short to minimize dissipation in the output transistors of the pulser. A projected graticule was used on the oscilloscope so that parallax could be eliminated from the experiment. Core E-6 was used for most of the ramp-F experiments. Cores I-4 and K-1 were tested with a ramp-F drive at various temperatures (Sec. III-B-4).

The measurement of $\dot{\phi}_p$, t_p and k is illustrated by Fig. 32. First, the peak in $\dot{\phi}$ was aligned with a vertical graticule line. Then, $\dot{\phi}_p$ and F_a were measured with a chopper and a voltage reference. Next, Δt and F_b were measured. The value of k was calculated from

$$k = \frac{F_b - F_a}{\Delta t} \tag{106}$$

and kt_p from

$$kt_p = F_a \quad . \tag{107}$$

For all except the low values of k, the $F(t)$ waveform had a nonlinearity near its beginning. The above method of measurement was used so that k and t_p would be determined from that portion of the ramp which existed during most of the switching prior to the peak in $\dot{\phi}(t)$. The linearity and slope of the ramp after the peak in $\dot{\phi}(t)$ is of no concern since it will not affect the determination of t_p and $\dot{\phi}_p$.

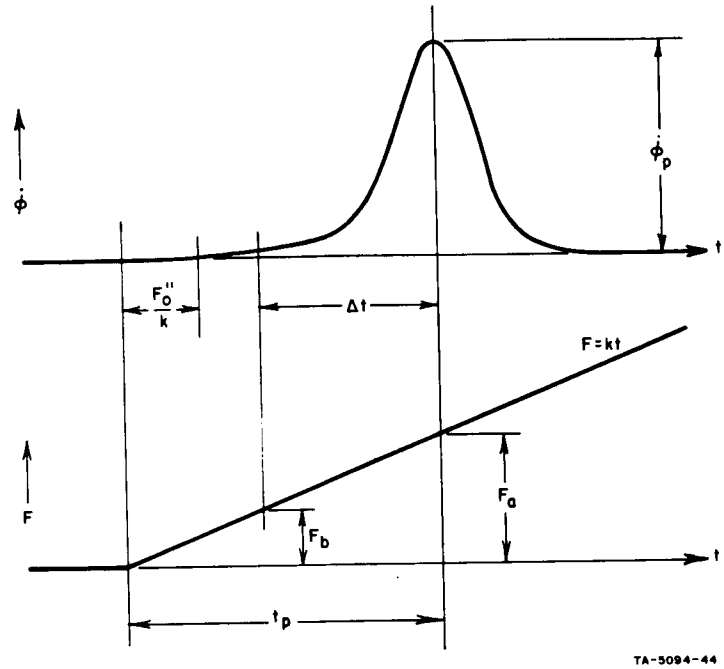


FIG. 32 DEFINITIONS OF MEASURED QUANTITIES FOR RAMP–F SWITCHING

The comparison of the experimental and computed $\dot{\phi}(t)$ waveform was done at a relatively low value of k (k = 0.837 amp-turn/μsec) to complement rather than duplicate the comparison made in Report 3 (k = 3.57 amp-turn/μsec). The experimental $\dot{\phi}(t)$ waveform and its corresponding $F(t)$ waveform are shown in Fig. 33. The numerically computed $\dot{\phi}(t)$ waveform is also included so that the shape of the waveforms can be compared. The values of λ_r and F''_{0r} used in the computation were determined from the $\dot{\phi}_p(k)$ and $kt_p(k)$ curves so that $\dot{\phi}_p$ and t_p would match the experimental values at k = 0.837. The values were λ_r = 0.0507 and F''_0 = 0.800 amp-turns. Note that once λ_r and F''_{0r} are correctly determined, very good agreement is obtained. The problem is one of determining how well a single set of values for λ_r and F''_{0r} [also ρ_{pr}, F_{Br}, and F_{0r} if two regions are used for $\dot{\phi}_p(F)$] will serve for all values of k.

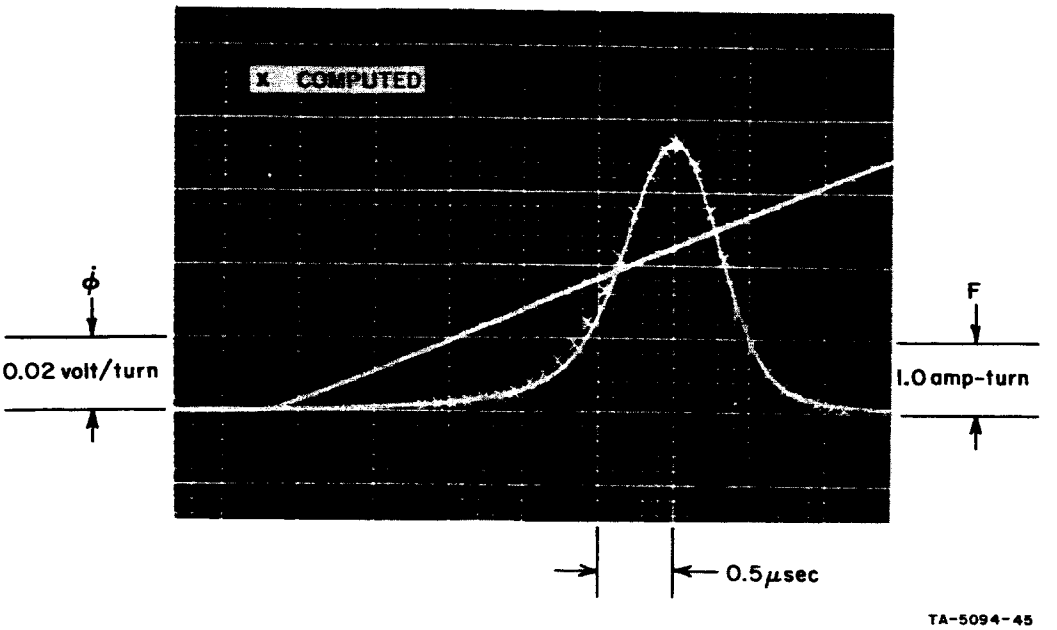


FIG. 33 EXPERIMENTAL vs. COMPUTED $\dot{\phi}(t)$ FOR RAMP F WITH λ_r AND F''_{0r} ADJUSTED TO MAKE $\dot{\phi}_p$ AND t_p AGREE
Core: E-6; Temp. = 30°C; k = 0.837 amp-turn/μsec; $\lambda_r = 0.049$; $F''_{0r} = 0.80$ amp-turn; $\nu = 1.30$

Experimental $\dot{\phi}_p(k)$ and $kt_p(k)$ points are included in Figs. 28 through 31. Note that the experimental $\dot{\phi}_p(k)$ data are below the numerically calculated curves (for step-F parameters) for all values of k. This is in agreement with the one point observation on p. 35 of Report 3. Thus λ needs to be decreased for all values of k.

Note that the experimental $\dot{\phi}_p(k)$ curve has a shape very similar to the numerically computed curve, *i.e.*, nearly a straight line on the log-log plot but with a slight downward curvature at low values of k. It appears that a vertical shift downward of the numerically computed curve would give good agreement for the entire curve. This vertical shift is obtained by varying λ as indicated by Eq. (104).

Strangely enough, the $kt_p(k)$ experimental and computed curves are in quite good agreement except for low values of k(*e.g.*, $k < 1$). This good agreement may be somewhat accidental, since a correction in λ to give agreement in $\dot{\phi}_p(k)$ will throw the $kt_p(k)$ curve out of agreement. Therefore, F''_0 has to be corrected to bring $kt_p(k)$ back into agreement [*cf.* Eqs. (103) and (104)]. This will be discussed in more detail in Part A-5.

5. Parameter Corrections

The values of λ and F_0'' need to be corrected to make the computed $\dot{\phi}_p(k)$ and $kt_p(k)$ curves agree with experimental ramp-F data. The corrected values can be calculated on the basis of Eqs. (103) and (104) and from the experimental and the numerically computed data. Define P as

$$P \equiv \frac{\dot{\phi}_p \text{ computed [using the } \phi_d \text{ parabolic model]}}{\dot{\phi}_p \text{ experimental}} \quad . \tag{108}$$

The value of ($\dot{\phi}_p$ computed) can be computed using step-F parameters, or any approximate values. Both values of $\dot{\phi}_p$ must be at the same value of k. Denote ramp-F corrected values by the subscript r. From the λ relationship of Eq. (104) we obtain

$$\lambda_r = \frac{\lambda}{P^{\nu+1}} \tag{109}$$

where λ is the value used in determining ($\dot{\phi}_p$ computed). To derive a corresponding equation for F_{0r}'' first solve Eq. (103) for F_0'' for ramp-F-corrected parameters. This results in

$$F_{0r}'' = kt_{p_{exp}} - A\left(\frac{k}{\lambda_r}\right)^{1/(\nu+1)} \tag{110}$$

where $t_{p_{exp}}$ is the experimental value. A is then also determined from Eq. (103) but using step-F values (or the approximate values) for λ and F_0'',

$$A = \left(kt_{p_{calc}} - F_0''\right)\left(\frac{\lambda}{k}\right)^{1/(\nu+1)} \tag{111}$$

where $t_{p_{calc}}$ is the calculated value of t_p. Substituting this equation into Eq. (110) and making use of Eq. (109) results in

$$F_{0r}'' = kt_{p_{exp}} - P(kt_{p_{calc}} - F_0'') \quad . \tag{112}$$

Equations (109) and (112) were used to get the following ramp-F values for Core E-6 at k = 6 amp-turns/μsec:

$$\lambda_r = 0.0507$$

$$F''_{0r} = 0.56 \text{ amp-turns} \quad .$$

The numerically computed $\dot{\phi}_p(k)$ and $kt_p(k)$ curves using this value of λ but the old value of F''_0 are included in Figs. 30 and 31. Note that the $\dot{\phi}_p(k)$ is in relatively good agreement for all k values, but that the original agreement of the $kt_p(k)$ curve has now been ruined. To correct this, F''_0 must be decreased. The calculation was repeated using both λ_r and F''_{0r}. The $\dot{\phi}_p(k)$ curve is included in Fig. 30. The $kt''_p(F)$ curve has been plotted in a new figure (Fig. 34) because the vertical axis contains F''_0. The $\dot{\phi}_p(k)$ curve has been altered primarily in the low-k region as a result of the change in F''_0. In this low-k region it has gone from slightly above to slightly below the experimental data. Hence, decreasing F''_0 increases the curvature of the $\dot{\phi}_p(k)$ curve if the ϕ_d model is used. This slight variation in curvature could be used to determine F''_0 very roughly. However, a value thus determined would be larger than the value 0.56 which is required to make t_p agree at k = 6 amp-turns/μsec. The computed $kt_p(k)$ curve (Fig. 34) now crosses the experimental curve at k = 6 where the value of F''_{0r} was determined. Unfortunately this $kt_p(k)$ curve cannot be made to agree for all k values. If ν is adjusted to make the $kt_p(k)$ curves agree for all k values, which would require a significant change in ν then the agreement in the $\dot{\phi}_p(k)$ curves would be ruined because of a change in the slope of the computed curve. Thus, adjusting ν would not help, but would considerably complicate matters.

So far in this discussion, very little has been said about determining ramp-F values for ρ_p, F_0, and F_B. In some cores the $\dot{\phi}_p(F)$ curve may require use of these parameters. It was noted, in preliminary computations, that a small deviation from the experimental $\dot{\phi}_p(k)$ curves at high k values could be traced to the inclusion of ρ_p, F_0 and F_B in the computer program. A careful examination of the experimental $\dot{\phi}_p(F)$ data for Core E-6 revealed that the $\dot{\phi}_p(F)$ curve was better described if ρ_p, F_0 and F_B were not introduced. If the experimental data had been extended to higher F values, perhaps ρ_p, F_0 and F_B would have been necessary to properly describe the $\dot{\phi}_p(F)$ curve. It was therefore impossible to make experimental observations on the effect of ramp-F switching on ρ_p, F_0 and F_B. It is possible,

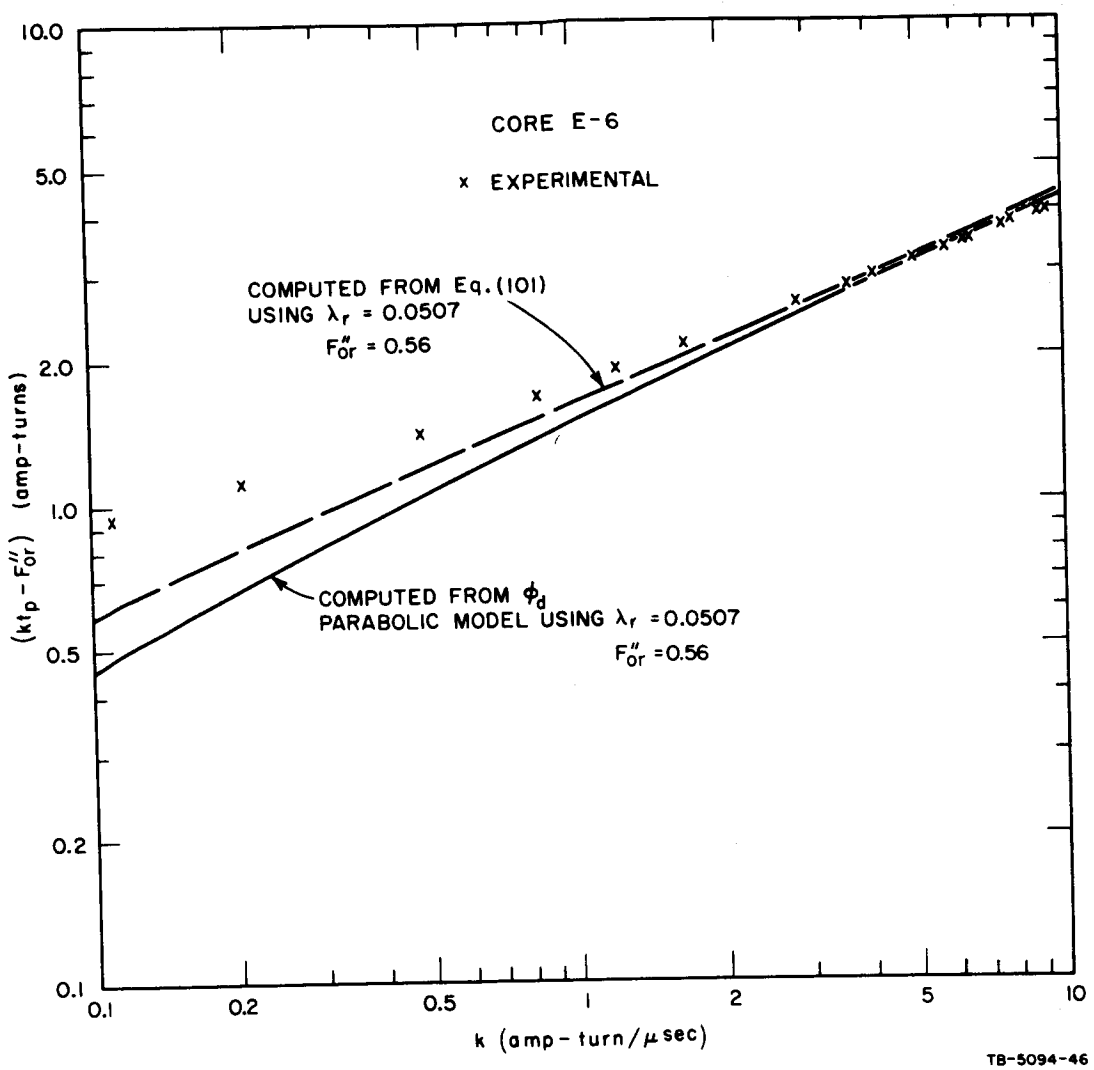


FIG. 34 $(kt_p - F''_{0r})$ vs. k FOR RAMP F USING λ_r AND F''_{0r}

however, to make some reasonable assumptions that will allow a determination of ramp-F values for ρ_p, F_0 and F_B (*i.e.*, ρ_{pr}, F_{0r}, and F_{Br}) from the values of λ_r and F''_{0r}. First, assume that the effect of ramp-F switching is to reduce the ordinate of the $\dot{\phi}_p(F)$ curve the same percentage for all F values. Then ρ_p will be reduced by the same factor as is λ. Therefore

$$\rho_{pr} = \rho_p \frac{\lambda_r}{\lambda} \quad . \tag{113}$$

Second, assume that the other effect of ramp-F switching is to shift the entire $\dot{\phi}_p(F)$ curve leftward. Then F_0 and F_B will be reduced by the same value as F_0''. Thus F_{0r} and F_{Br} are given by

$$F_{0r} = F_0 - (F_0'' - F_{0r}'') \tag{114}$$

and

$$F_{Br} = F_B - (F'' - F_{0r}'') \quad . \tag{115}$$

The equations of continuity of the $\dot{\phi}_p(F)$ curve as given on pp. 17 and 18 of Report 3 are preserved by these assumptions. It should be emphasized that Eqs. (113), (114), and (115) have no experimental verification.

The physical mechanisms responsible for the effects of ramp-F switching are not presently understood. They are probably closely related to the effects of partial setting (*cf.* p. 129 of Report 3). It is not surprising that good agreement cannot be obtained for all values of k by adjusting only λ and F_0''. See, for example, the effects that partial setting has on t_p as observed in Figs. 34(a) and 35 of Report 3, and also on the static $\phi(F)$ curves as observed in Figs. 30 of Report 3. Similar effects may also occur in ramp-F switching. This needs further investigation.

In practical applications where $F(t)$ can be approximated by a ramp, it is certainly better to make an adjustment in the value of λ and F_0'' rather than ignore the effects of ramp-F switching altogether. If a new value for F_0'' is determined at a value of k in the middle of the range of interest, quite good results should be obtained. The experimental determination of λ_r and F_{0r}'' can be done with measurements at only a few k values if the results are plotted with log-log scales, since nearly straight lines result.

When only ramp-F values of parameters are needed, the $\dot{\phi}_p(F)$ measurement can be eliminated and λ_r, F_{0r}'' and ν determined directly from experimental $\dot{\phi}_p(k)$ and $kt_p(k)$ curves. First, a straight line is drawn tangent to the log-log $\dot{\phi}_p(k)$ curve at the high-k end. From the slope of this straight line, ν can be determined according to Eq. (104). Next, λ can be calculated from one point on this straight line by using Eq. (102). Finally, F_0'' can be determined from an appropriate point of the experimental $kt_p(k)$ data by using Eq. (101). These approximate values of λ and F_0'' can then be used to compute $\dot{\phi}_p(k)$ and $t_p(k)$ using the ϕ_d parabolic model.

These computed curves, together with Eqs. (109) and (112) can be used to determine final values of λ_r and F''_{0r}.

6. Summary

The parabolic switching model

$$\dot{\phi} = \lambda(F - F''_0)^{\nu}\left(1 - \frac{\phi^2}{\phi_s^2}\right)$$

was used for ramp-F drive to derive equations for $\dot{\phi}_p(k)$ and $t_p(k)$, where k is the slope of the ramp. It was shown that $\dot{\phi}_p$ is independent of F''_0 for this model. These equations are useful as an intuitive aid for analyzing the experimental results, and in determining corrected values for λ and F''_0 for ramp-F switching. The numerical computation of $\dot{\phi}_p(k)$, and $t_p(k)$ with the ϕ_d parabolic model and a digital computer was discussed and the results compared to those obtained analytically with the simpler form of the parabolic model given above. The values were nearly equal for high values of k (*e.g.*, $\dot{\phi}_p$ values differed by 4.3 percent at k = 10 amp-turn/μsec) but differed considerably at low values of k. The details of the experimental measurements were discussed and the $\dot{\phi}_p(k)$ and $t_p(k)$ curves for Core E-6 were compared to the computed curves using step-F parameters. The experimental $\dot{\phi}_p(k)$ curve was significantly lower (*e.g.*, 15 percent at k = 1 amp-turn/μsec). The experimental $t_p(k)$ curve was in close agreement with the computed curve except for low values of k. The algebraic equations for $\dot{\phi}_p(k)$ and $t_p(k)$ were used to derive simple relationships for computing ramp-F corrected values for λ and F''_0. These values were used in the ϕ_d parabolic model to again compute $\dot{\phi}_p(k)$ and $t_p(k)$. The $\dot{\phi}_p(k)$ curves were now in good agreement for all k values. The $t_p(k)$ curves were inexact agreement at only one k value, which is the point at which the two curves crossed each other. The agreement in the vicinity of this crossing point was good enough for many practical applications.

B. EFFECTS OF TEMPERATURE

1. Introduction

In a practical magnetic circuit, operation is influenced by variations in temperature. Analysis of these circuits by use of a switching model therefore requires a knowledge of the variation of the core parameters *vs.*

temperature. It is also necessary to verify that the model which has been used at room temperature is also valid for a wide temperature range. In addition, it is helpful for qualitative studies of circuit operation to know the general trends and approximate magnitude of the effects of varying temperature.

The effects of temperature variations were determined by measuring the core properties at each of several temperatures from −50°C to +75°C. First, the static $\phi(F)$ curves were measured, starting from both a hard remanent state and a partially set state. Second, the $\dot{\phi}_p(F)$ curve for step-F switching was measured at each temperature. Third, the $\dot{\phi}_p(k)$ and $kt_p(k)$ curves for ramp-F switching were measured at each temperature. The core parameters were determined from these various curves and plotted *vs.* temperature.

These temperature effects were measured for two cores, Core I-4, and Core K-1. The dimensions of these cores are given in Table II. Core I-4 is a thin ring which was ultrasonically cut from a disc of Indiana General 5209 ferrite. It is the same core used in Report 3 for studying the effects of partial setting. The disc from which Core I-4 was cut was one of a batch of 10 discs which were magnetically tested for uniformity. Core K-1 is a Lockheed 100SC1 switch core. It is from a group of six cores which were magnetically tested for uniformity.

The temperature of the core was controlled by a commercial temperature test chamber which used electrical heating and expansion of CO_2 for cooling. The inaccuracy of the temperature is less than ±1.5°C at all temperatures.

2. Static $\phi(F)$

The static $\phi(F)$ curves were measured starting from both a hard remanent state and a partially set state for each temperature. The partially set state was obtained by switching the core from $-\phi_r$ to $\phi = 0$ by a 1-μsec rectangular pulse. The static $\phi(F)$ curve from this partially set state was then measured for both a positive and a negative polarity of F.

The model for the static $\phi(F)$ curves which we have been using (see pp. 3-7 and pp. 74, 75 of Report 2) was fitted to each hard-state experimental static $\phi(F)$ curve. The values of the parameters thereby determined were plotted *vs.* temperature. However, this $\phi(F)$ model is not applicable for partially set $\phi(F)$ curves. An appropriate model has not yet been

Table II

DIMENSIONS, SWITCHING PARAMETERS, AND TEMPERATURE COEFFICIENTS OF CORES I-4 AND K-1

	CORE I-4	CORE K-1
	Indiana General 5209 Ultrasonically Cut 30 ± 0.5°C	Lockheed 100SC1 Commercial 24.6 ± 0.5°C
DIMENSION		
r_0 (mm)	3.78	1.27 (50 mils)
r_i (mm)	3.43	0.89 (35 mils)
r_0/r_i	1.10	1.43
h (mm)	0.848	0.76 (30 mils)
w (mm)	0.35	0.38
l_i (mm)	21.55	5.59
l_o (mm)	23.77	7.98
PARAMETER		
ϕ_r (maxwells)	5.97	6.43
ϕ_s (maxwells)	6.55	7.10
H_a (amp-turns/meter)	230.0	200.0
H_q (amp-turns/meter)	43.1	37.3
H_n (amp-turns/meter)	37.0	31.4
λ (ohm/turn$^{\nu+1}$amp$^{\nu-1}$)	0.134	0.64
F_0'' (amp-turn)	1.28	0.35
ν	1.19	1.21
λ_r (ohm/turn$^{\nu+1}$amp$^{\nu-1}$)	0.119	0.477
TEMPERATURE COEFFICIENT		
$\alpha_{\phi r}$ (1/°C)	-0.0030	-0.003o
$\alpha_{\phi s}$ (1/°C)	-0.0025	-0.0035
α_{Hq} (1/°C)	-0.0077	-0.0095
α_{Hn} (1/°C)	-0.0095	-0.0097
α_λ (1/°C)	-0.001	+0.0017
$\alpha_{F_0''}$ (1/°C)	-0.0084	-0.0065
α_ν (1/°C)	+0.0008	-0.0002
$\alpha_{\lambda r}$ (1/°C)	-0.003	+0.001

obtained for partially set states. Therefore, the effects of temperature upon these partially set $\phi(F)$ curves will be judged by looking at the curves themselves.

The pulse sequence consisted of five rectangular pulses: a positive PARTIAL-SET pulse, a positive or negative TEST pulse, a negative CLEAR pulse, a positive CLEAR pulse, and finally another negative CLEAR pulse (see p. 84 of Report 3). The width of the SET pulse was 6 msec. The first CLEAR pulse followed closely the end of the SET pulse. The flux switched by the PARTIAL-SET and the TEST pulses was measured at the time of the first CLEAR pulse by means of a flux reference (see Appendix F of Report 2). The peak flux, not remanent flux, was measured for all the $\phi(F)$ curves of this report. The negative CLEAR pulses were greater than 15 amp-turns for Core I-4 and 7.0 amp-turns for Core K-1 (duration = 10 μsec).

The general effects of temperature on the hard-state static $\phi(F)$ is shown in Fig. 35 for Core I-4 and Fig. 36 for Core K-1. It was found that the $\phi(F)$ model could be fit quite well for all the temperatures for Core I-4 (*e.g.*, less than 3 percent error except right at the threshold

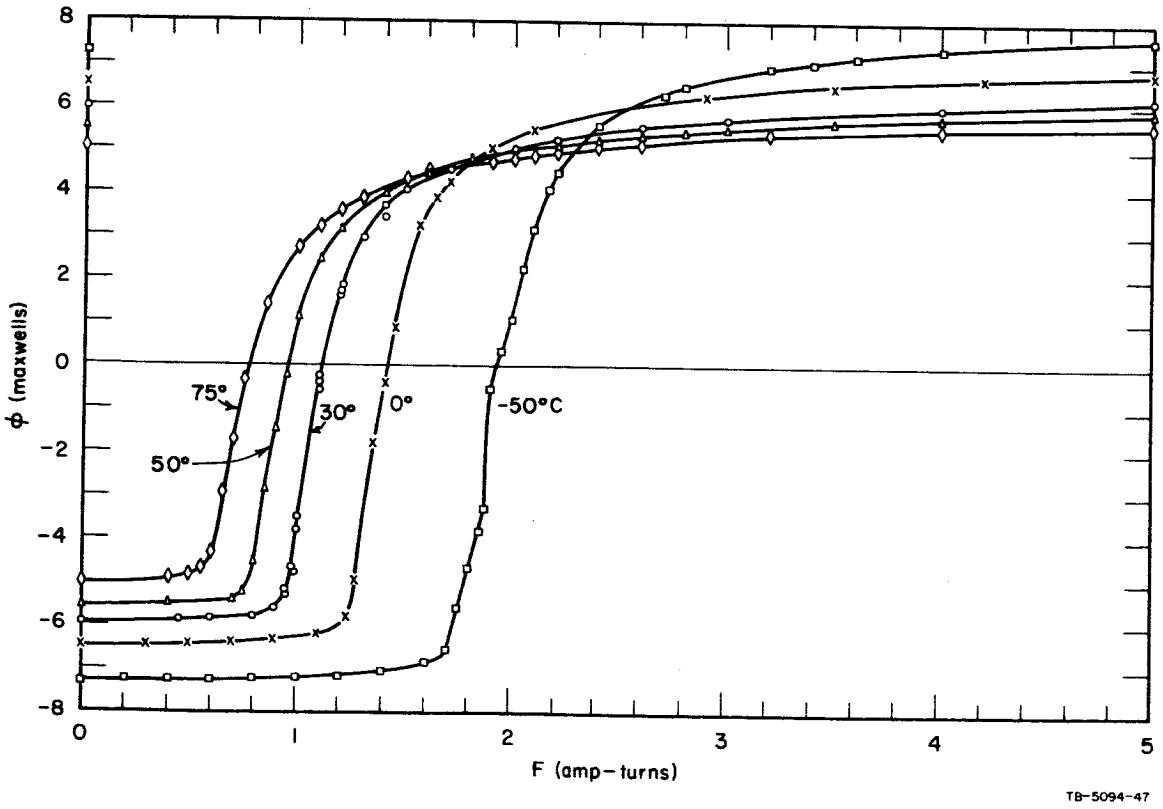


FIG. 35 STATIC ϕ(F) CURVES vs. TEMPERATURE OF CORE I-4

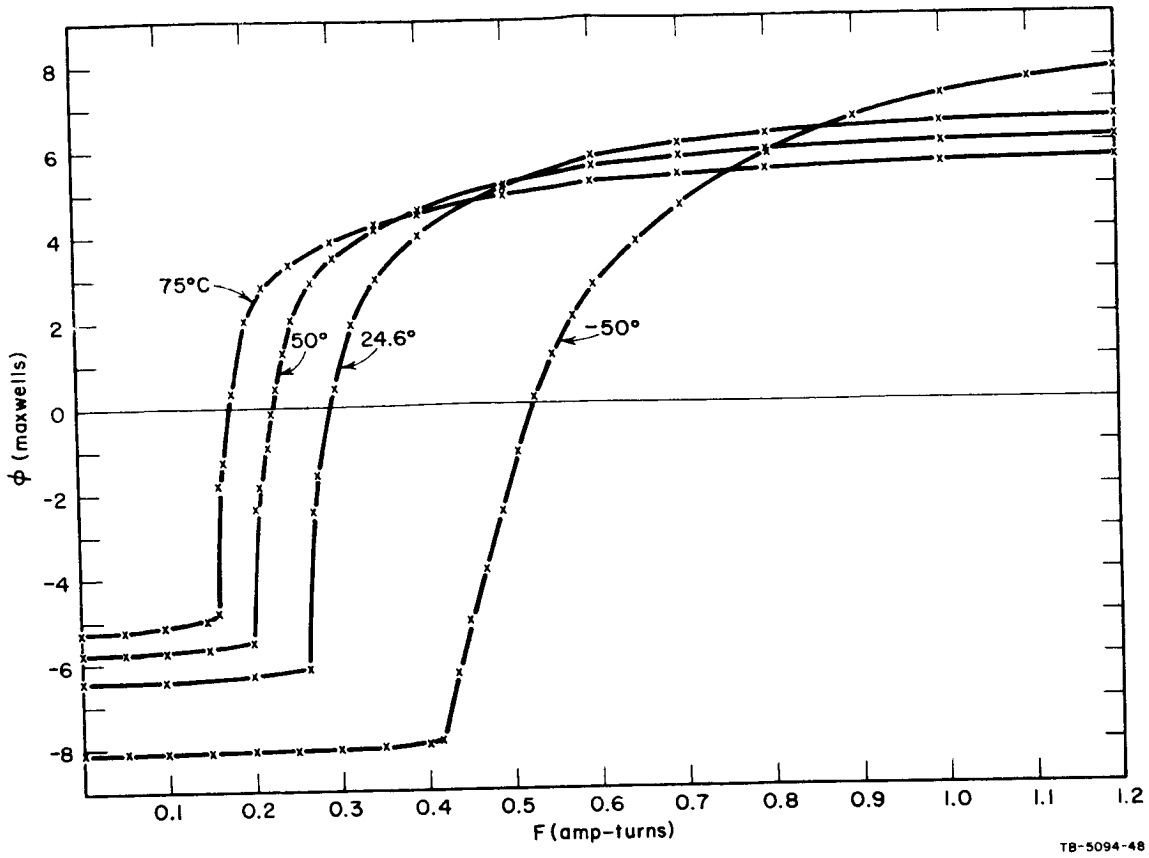


FIG. 36 STATIC $\phi(F)$ CURVES vs. TEMPERATURE OF CORE K-1

where the experimental curve has a more rounded corner). The irregularities in the -50°C curve for Core I-4 are, of course, smoothed out in the computed curve, but the percentage error due to these irregularities is less than 3 percent. Experimental $\phi_\epsilon(F)$ data was not included in these measurements. Therefore, H_a can only be very roughly determined by using the portion of the $\phi(F)$ curves between $F = 0$ and $F = F_{th}$. The value of H_a previously determined at room temperature was adequate for all temperatures except -50°C for both cores. At -50°C H_a had to be drastically increased (*e.g.*, from about 200 to 900 amp-turns) to prevent the computed curve from rising above the experimental curve for $0 < F < F_{th}$. This increase in H_a at lower temperatures means that the $\phi(F)$ curves at $-\phi_r$ are flatter at lower temperatures. This is qualitatively consistent with permeability measurements which generally show an increase in permeability as temperature increases and measurements of squareness ratio which decreases with an increase in temperature. The increase in ϕ_r, ϕ_s, and F_c as temperature decreases are well known effects.

The static $\phi(F)$ curves of Core K-1 cannot be as well described by the model as those for Core I-4. The experimental $\phi(F)$ curves have a relatively steep section just above the threshold, which the computed curve does not exhibit. Thus the computed curve has a lower F_{th} than does the experimental curve. Perhaps the inner radius of this type of core has a higher H threshold than the parts of the core with larger radius. The static $\phi(F)$ model assumes uniform material properties throughout the volume of the core. In practice, this problem is not serious because the threshold in the switching model is determined by F_0'', which is generally above F_{th} anyway. In other respects these $\phi(F)$ curves were quite well described by the model. The worst error, not including the threshold problem, is about 7.5 percent, and that is over a very small range of F near $F = 0.21$ amp-turns for the 75°C curve.

The values of H_q, H_n, ϕ_r and ϕ_s are plotted in Figs. 37 and 38 for Cores I-4 and K-1, respectively. These values were used in the computations of $\dot{\phi}_p(k)$ for a ramp-F drive in Part B-4. Since these curves are fairly linear, a temperature coefficient can be used to correct the parameters for variations in temperature. The values of these temperature coefficients were determined at 30°C and are given in Table II. These values are the primary objective of these measurements. It appears as if H_n for Core I-4 might go to zero at a temperature somewhat above 100°C but below the Curie temperature. This is possible since H_n is merely the vertical asymptote for the hyperbolic static $B(H)$ curve. The ratio ϕ_r/ϕ_s is nearly constant at 0.90 for Core K-1, but for Core I-4 has a maximum of 0.92 at about 30°C and a lower value above and below this temperature. At 75°C it is 0.86. The sharpness of the wing of the static $B(H)$ curve can be judged by the ratio H_q/H_n which is ≥ 1. For H_q/H_n near unity, a very sharp wing is obtained and the side of the $B(H)$ curve is very steep. This is usually a desirable characteristic. As H_q/H_n increases the wing becomes more rounded and the side less steep. This H_q/H_n ratio is shown in Fig. 39. It must be remembered that Core K-1 has a steep $\phi(F)$ curve just above F_{th} which is not properly described by the static $\phi(F)$ model. Therefore Core K-1 is actually a little better than indicated by the ratio H_q/H_n as far as steepness is concerned. The core geometry has been accounted for in determining H_q and H_n so that H_q and H_n are a measure of the average material properties rather than the core properties.

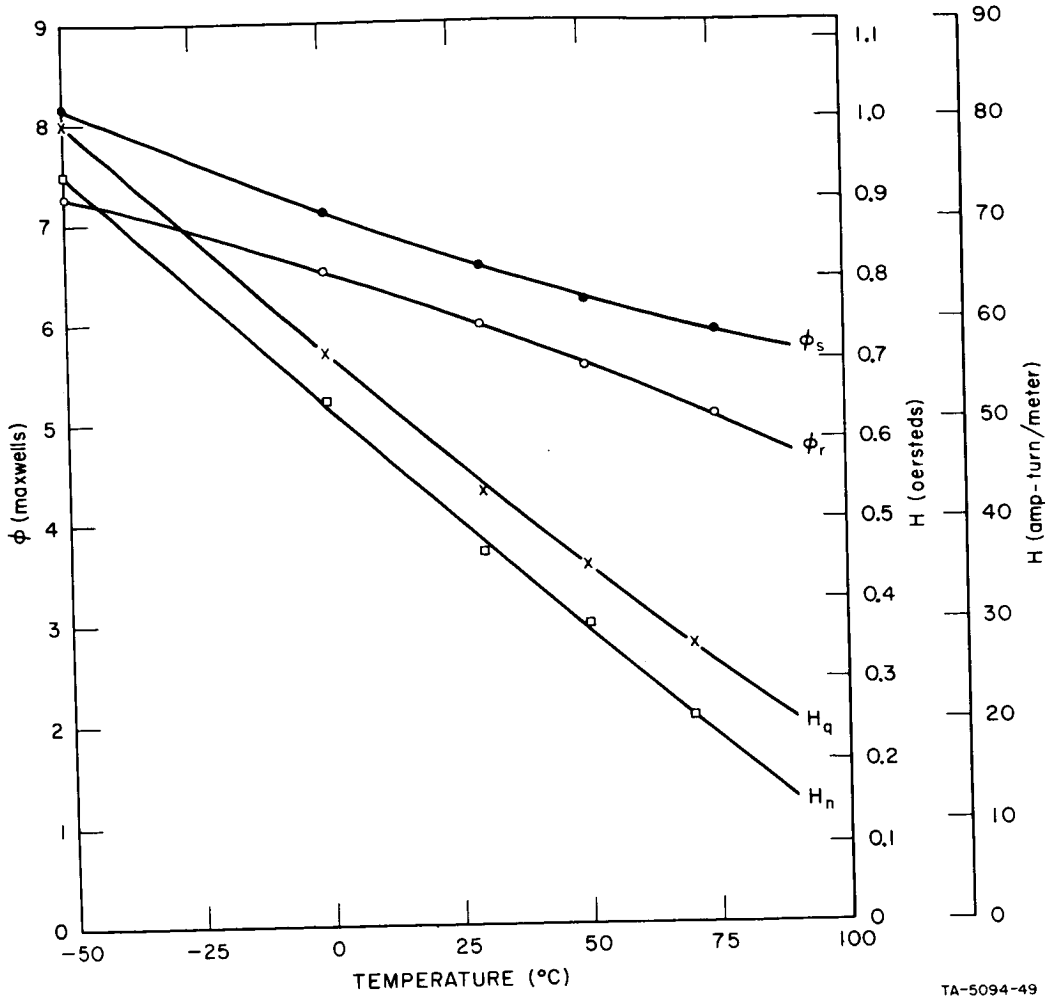


FIG. 37 STATIC ϕ(F) PARAMETERS vs. TEMPERATURE FOR CORE 1-4

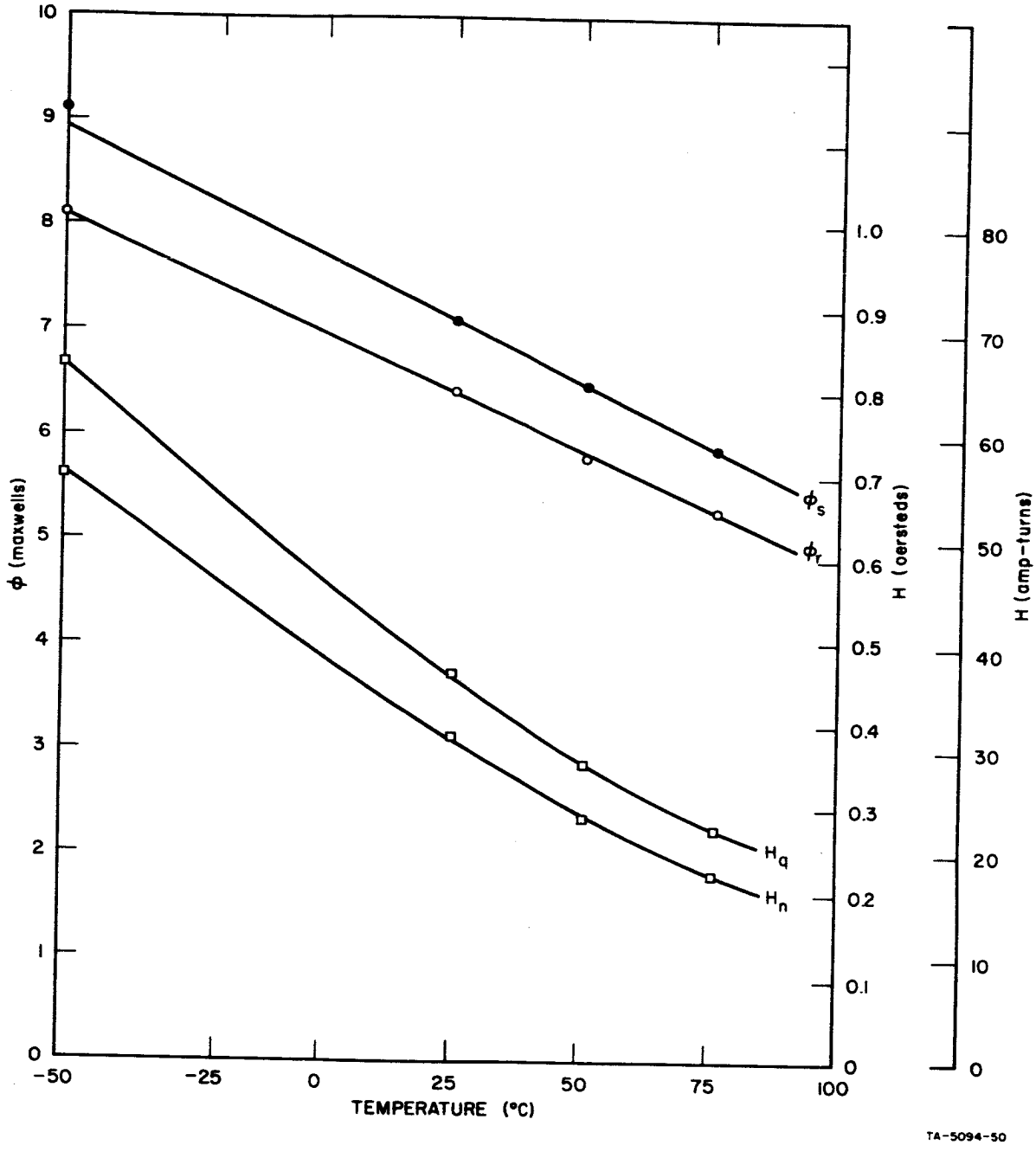


FIG. 38 STATIC ϕ(F) PARAMETERS vs. TEMPERATURE FOR CORE K-1

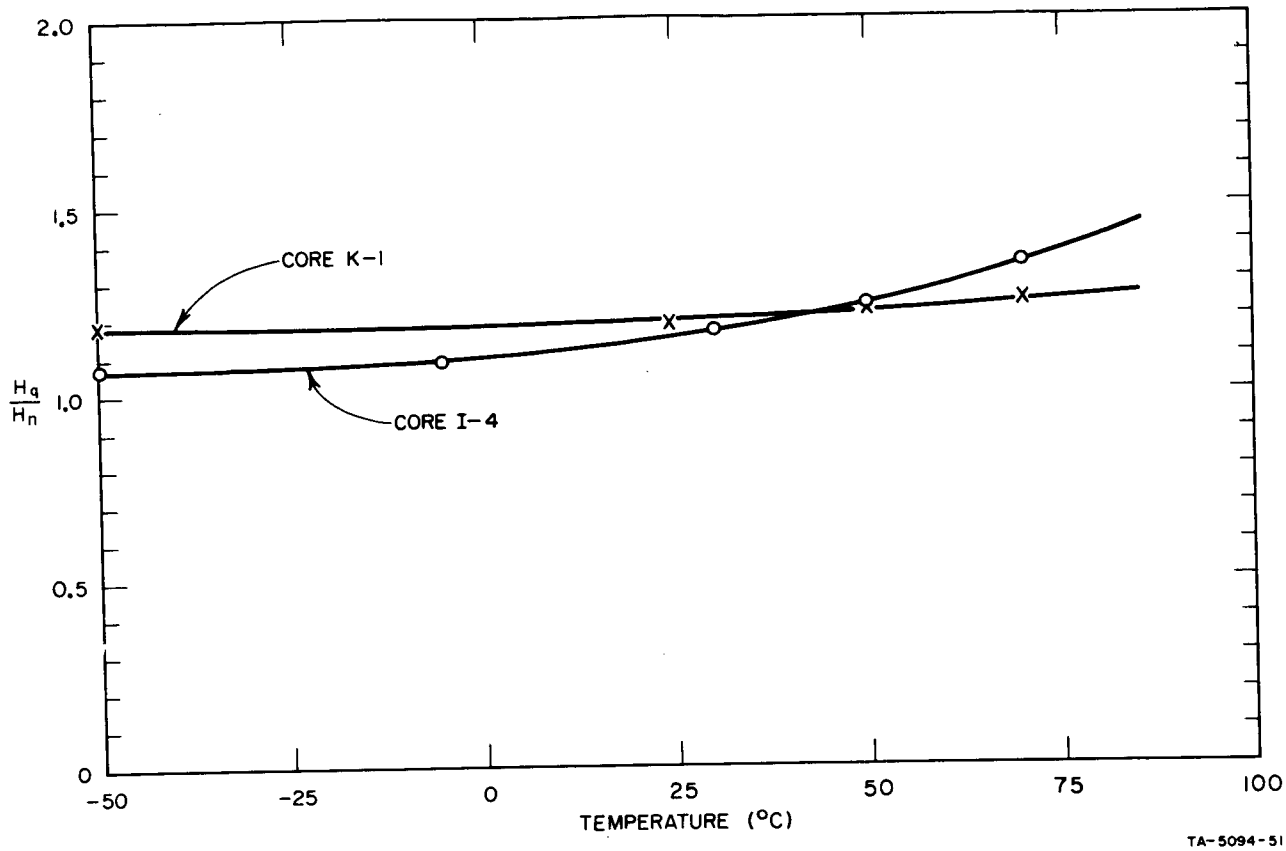


FIG. 39 WING SHARPNESS OF STATIC $\phi(F)$ CURVES vs. TEMPERATURE

The static $\phi(F)$ curves for partially set states were obtained with a 1-μsec PARTIAL-SET pulse except for Core I-4 at -50°C. For this case, the threshold had increased considerably and the current driver used for partial setting was unable to switch the core to $\phi = 0$ in 1 μsec. Therefore, the width had to be increased to 1.6 μsec. In general, the amplitude of the 1-μsec PARTIAL-SET pulse was adjusted so that the core would be partially set from $\phi = -\phi_r$ to $\phi = 0$. The static $\phi(F)$ curve was obtained for both $+F$ and $-F$. The curve for $-F$ will be shown in the first rather than the fourth quadrant for easy comparison with the $+F$ curves.

The static $\phi(F)$ curves for $+F$ and $-F$ for various temperatures are shown in Figs. 40 and 41, respectively for Core I-4 and in Figs. 42 and 43 for Core K-1. Qualitatively, the general character of these partially set curves is preserved throughout the temperature range. Note that the wing of each $+F$ curve is more rounded than the threshold. This characteristic is reversed for the $-F$ curves, where the threshold is more rounded than the wing. This property is enhanced for very slow partial setting. See, for

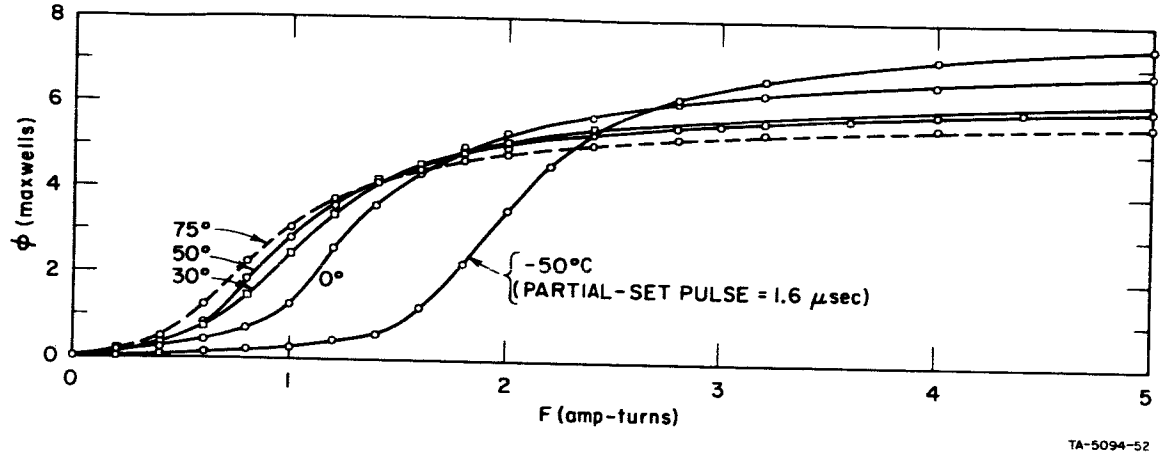


FIG. 40 STATIC $\phi(+F)$ vs. TEMPERATURE OF PARTIALLY-SET CORE 1-4
PARTIAL-SET pulse = 1 μsec (except at 50°C)

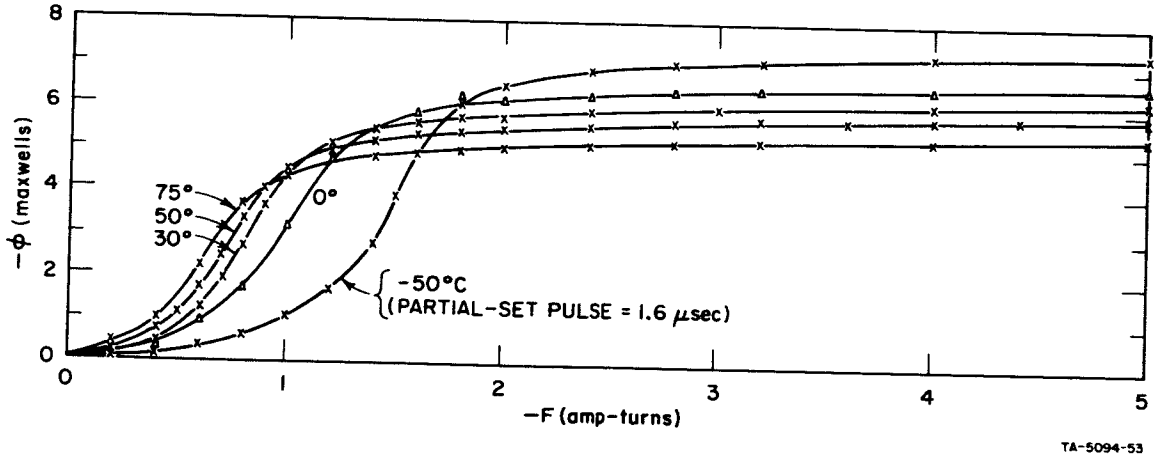


FIG. 41 STATIC $\phi(-F)$ vs. TEMPERATURE OF PARTIALLY-SET CORE 1-4
PARTIAL-SET pulse = 1 μsec (except at -50°C)

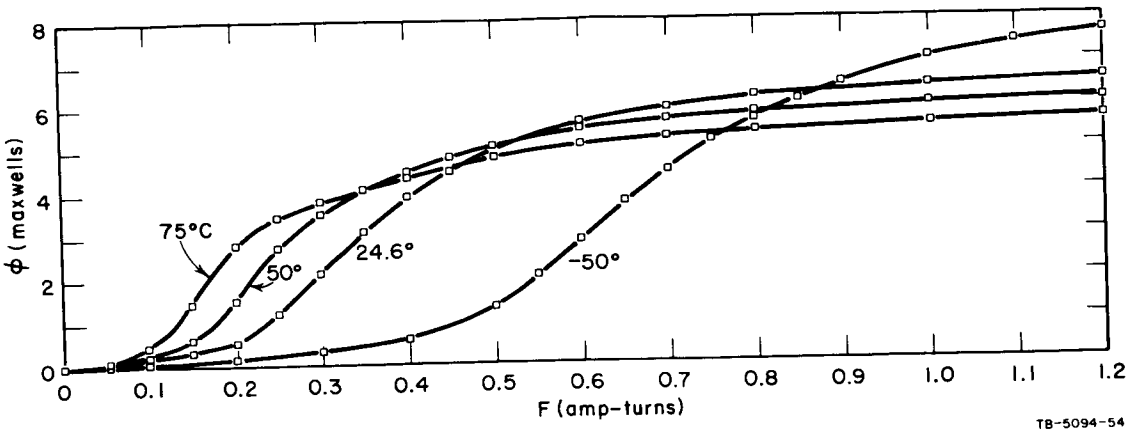


FIG. 42 STATIC $\phi(+F)$ vs. TEMPERATURE OF PARTIALLY-SET CORE K-1
PARTIAL-SET pulse = 1 μsec

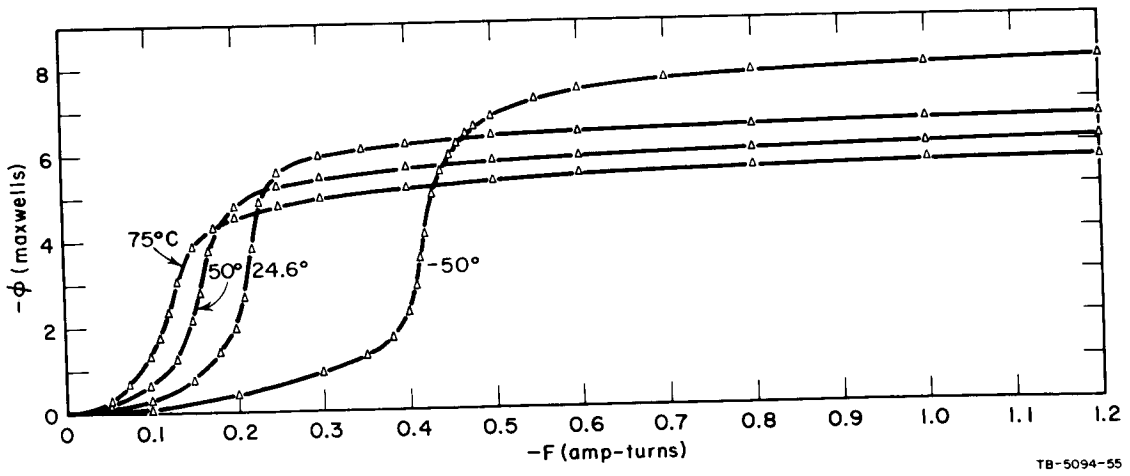


FIG. 43 STATIC $\phi(-F)$ vs. TEMPERATURE OF PARTIALLY-SET CORE K-1
PARTIAL-SET pulse = 1 μsec

example, Figs. 30(c) and (d) on page 91 of Report 3. The physical reason for this property can be explained as follows. During very slow partial setting, the regions of the core which have a lower H_c switch rapidly and higher H_c regions switch more slowly. When switching is continued in the case of a positive TEST pulse the remaining unswitched regions are mostly of a high H_c which tends to give a relatively long extensive wing in the $\phi(F)$ curve as all the high H_c regions are finally switched at high F. In contrast, when the F is negative (*i.e.*, opposite in polarity to the PARTIAL-SET pulse) the TEST pulse switches the partially set flux back again to $-\phi_r$. Since this involves mostly the lower threshold regions a relatively sharp wing is obtained because few of the higher threshold regions had been partially set. This physical explanation is supported by the fact that this property is most evident for very slow partial setting. Fast partial setting switches all regions of the core nearly equally, thereby reducing the differences between the $+F$ and $-F$ curves of $\phi(F)$. A better comparison of the positive and negative partially set curves and the major static $\phi(F)$ curves can be made in Fig. 44 for Core I-4 and in Fig. 45 for Core K-1. These figures include curves for one high and

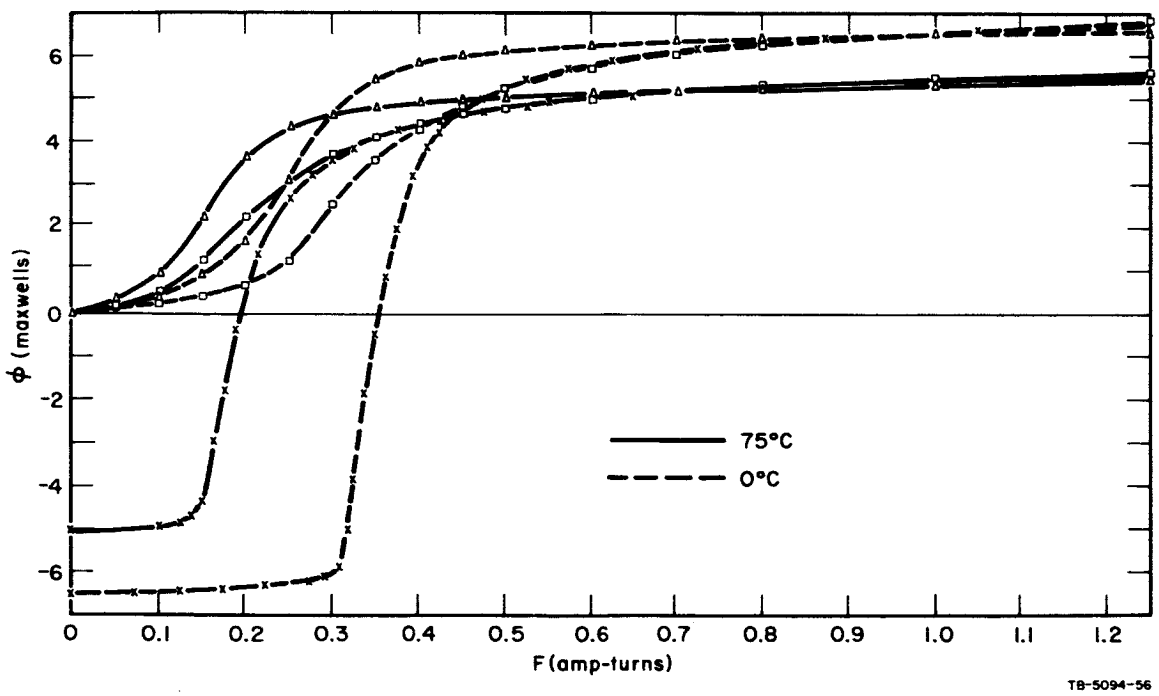


FIG. 44 MAJOR AND PARTIALLY-SET STATIC ϕ(F) OF CORE I-4 AT 75° AND 0°C
PARTIAL-SET pulse = 1 μsec

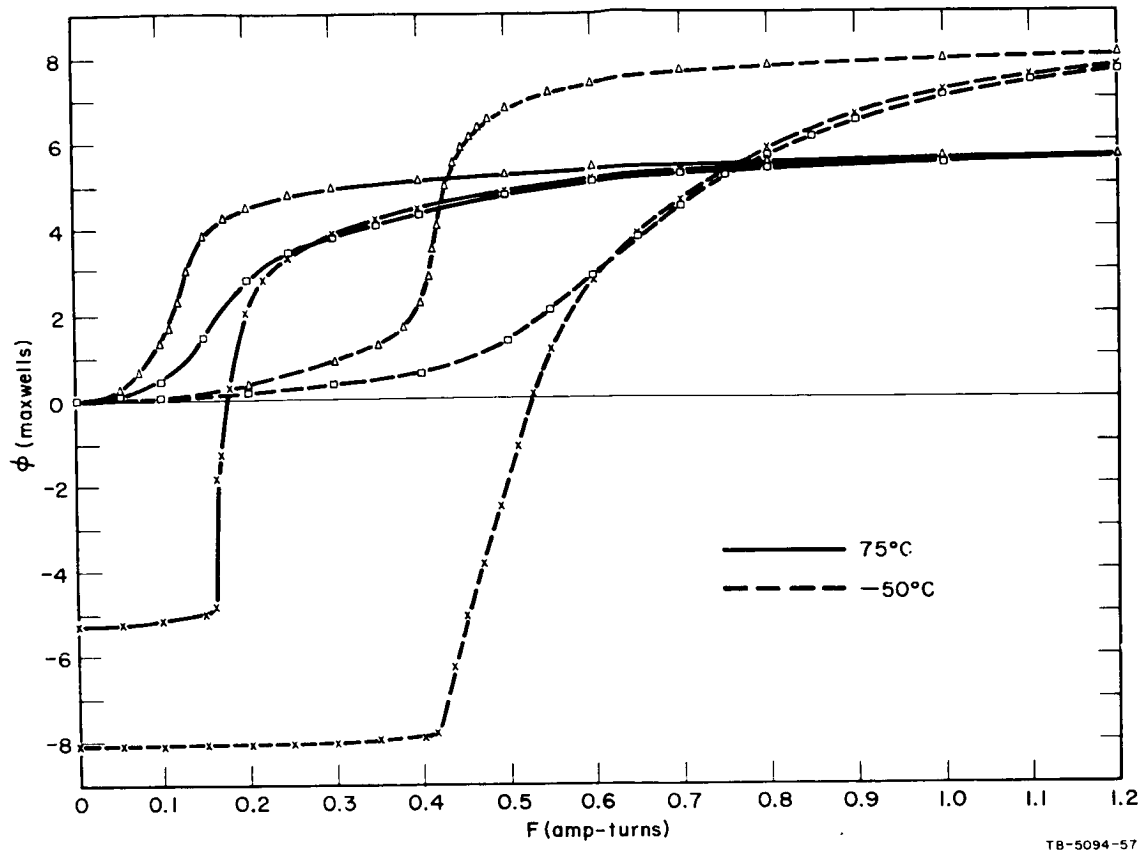


FIG. 45 MAJOR AND PARTIALLY-SET STATIC $\phi(F)$ OF CORE K-1 AT 75° AND -50°C
PARTIAL-SET pulse = 1 μsec

one low temperature (75°C and 0°C for Core I-4 and 75°C and -50°C for Core K-1). The 0°C curve was used rather than the -50°C curve for Core I-4 to avoid the one case in which a 1.6 μsec-duration PARTIAL-SET pulse was used. The effects of partial setting upon the static $\phi(F)$ curves seems to be very much the same for the entire temperature range.

3. Step-F Switching

The effects of temperature on step-F switching will be studied by means of the $\dot{\phi}_p(F)$ curves. The primary goal is to obtain temperature coefficients for λ, F_0'', and ν. Experimental $\dot{\phi}_p(F)$ data were measured for Core I-4 and Core K-1 at each of several temperatures in the range -50°C to +75°C. These data were used to determine λ, F_0'' and ν versus temperature. No attempt was made to determine ρ_p, F_0 and F_B versus temperature because these values cannot be accurately determined for the range of F used

for the two cores tested. No $\dot{\phi}_p(F)$ *vs.* temperature curves were taken for partially set states. This should also be investigated in the future.

The pulse sequence consisted of a SET pulse followed by a negative CLEAR pulse, a positive CLEAR pulse, and another negative CLEAR pulse. This CLEAR pulse sequence is used to completely remove all history effects (see p. 83 of Report 3). The SET pulse had a 60-nsec rise time (10% to 90% points) and a maximum amplitude of 5 amp-turns. It was supplied by four Digital Equipment Corp. Model 62 current drivers in parallel. The negative clear pulses were greater than 15 amp-turns for Core I-4 and 7.0 amp-turns for Core K-1. The $\dot{\phi}_p(F)$ and $\dot{\phi}_p(k)$ data for Core I-4 at 30°C were taken before the core had been cooled to −50°C. The other data were taken afterward. Dropping the temperature to −50°C and returning to room temperature seemed to alter the core's magnetic properties a small amount. Thus the 30°C $\dot{\phi}_p(F)$ and $\dot{\phi}_p(k)$ curves are not completely consistent with the rest of the data and have therefore been omitted. The exact cause of this small effect has not been determined. The static $\phi(F)$ curves at 30°C were taken both before and after the first −50°C excursion.

The $\dot{\phi}_p(F)$ curves for different temperatures are shown in Fig. 46 for Core I-4 and in Fig. 47 for Core K-1. It is readily apparent in these figures that the only major effect on $\dot{\phi}_p(F)$ of increasing temperature is the decrease in F_0''. The decrease in F_0'' is to be expected since it is commonly known that H_c decreases as temperature increases. This is illustrated by the static $\phi(F)$ curves of Figs. 35 and 36. The values of F_0'', λ, and ν were determined for each temperature by plotting the $\dot{\phi}_p(F)$ data on log-log paper. These values are shown *vs.* temperature in Fig. 48 for Core I-4 and in Fig. 49 for Core K-1. Note that ν is nearly constant with temperature. The value of λ is also nearly constant for Core I-4, but increases somewhat with temperature for Core K-1. The value of λ depends upon a number of physical quantities so that it is not surprising if its temperature behavior differs for different cores. If we compare the parabolic model to the model of Menyuk and Goodenough,[14] then λ will be determined by the following quantities: the viscous damping parameter β, the square of the saturation magnetization, M_S^2, the density of domain nucleations, and $\langle\cos\theta\rangle$ which accounts for the statistical variation in the directions of the easy axes of individual ferrite grains.

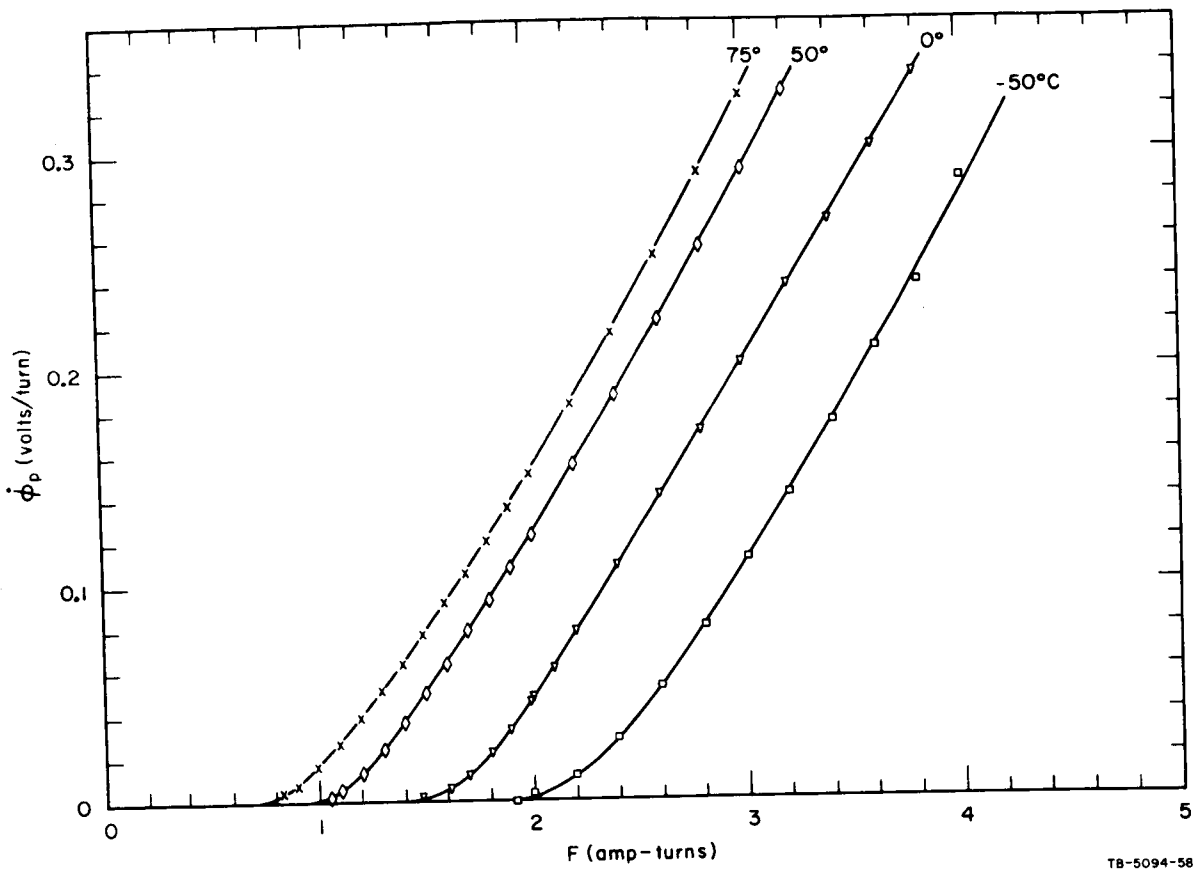


FIG. 46 $\dot{\phi}_p(F)$ vs. TEMPERATURE OF CORE 1-4

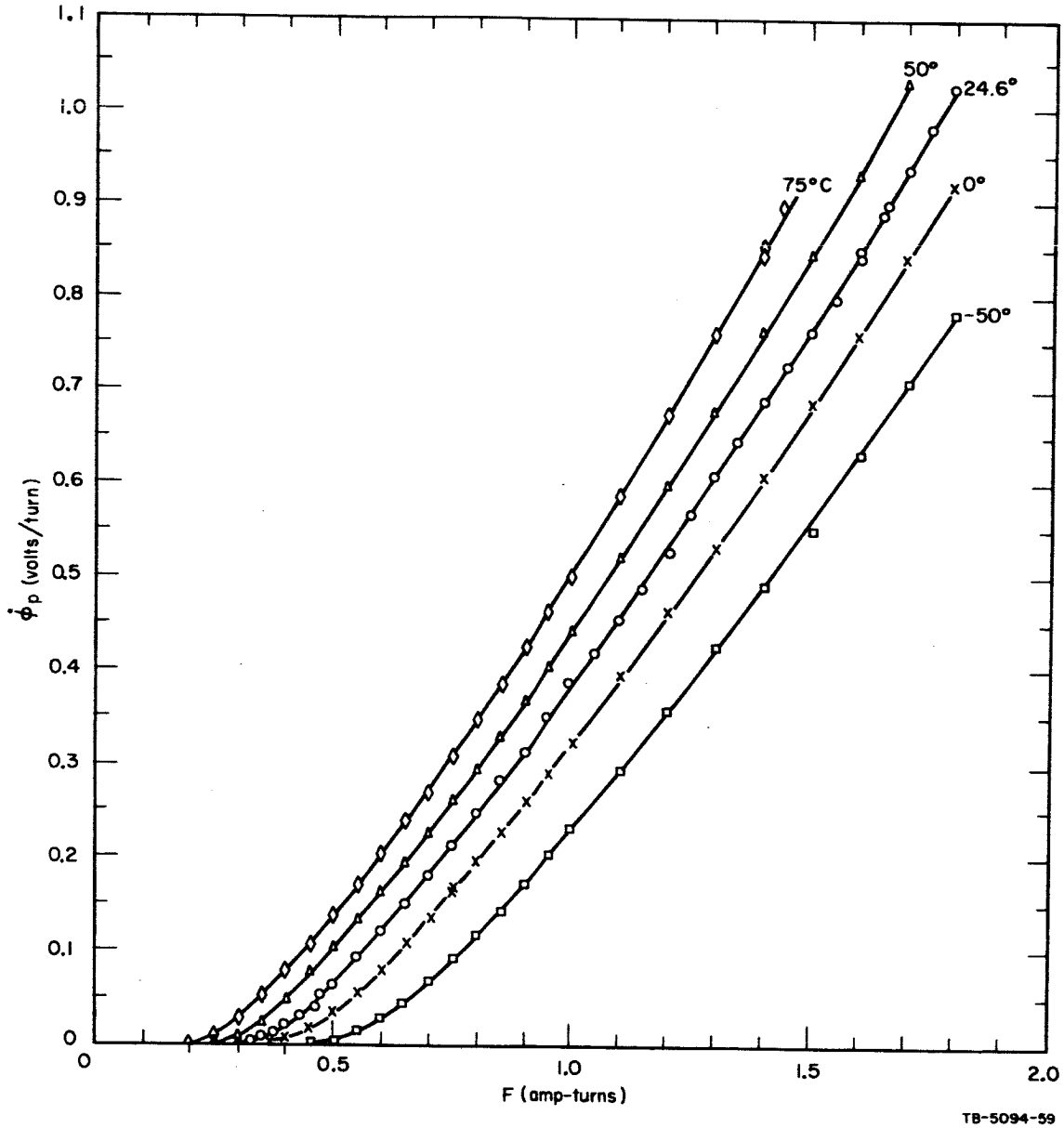


FIG. 47 $\dot{\phi}_p(F)$ vs. TEMPERATURE OF CORE K-1

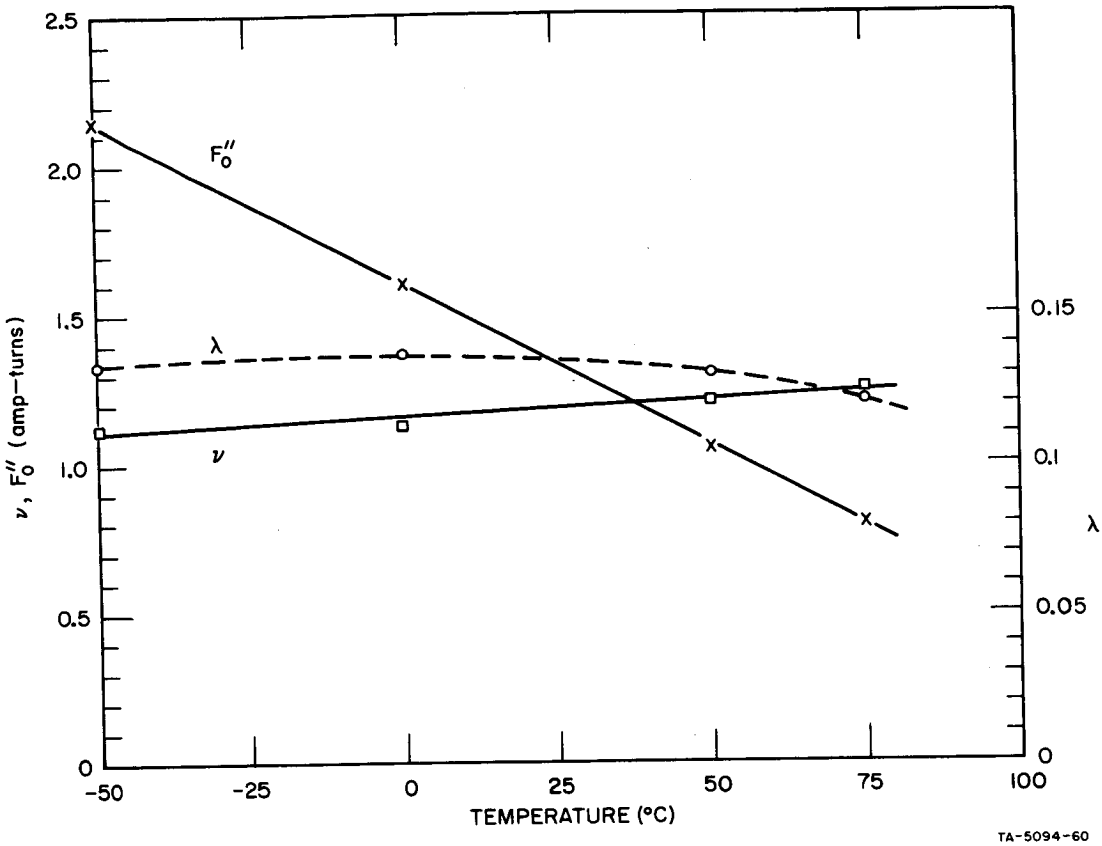


FIG. 48 $\dot{\phi}_p(F)$ PARAMETERS vs. TEMPERATURE FOR CORE 1-4

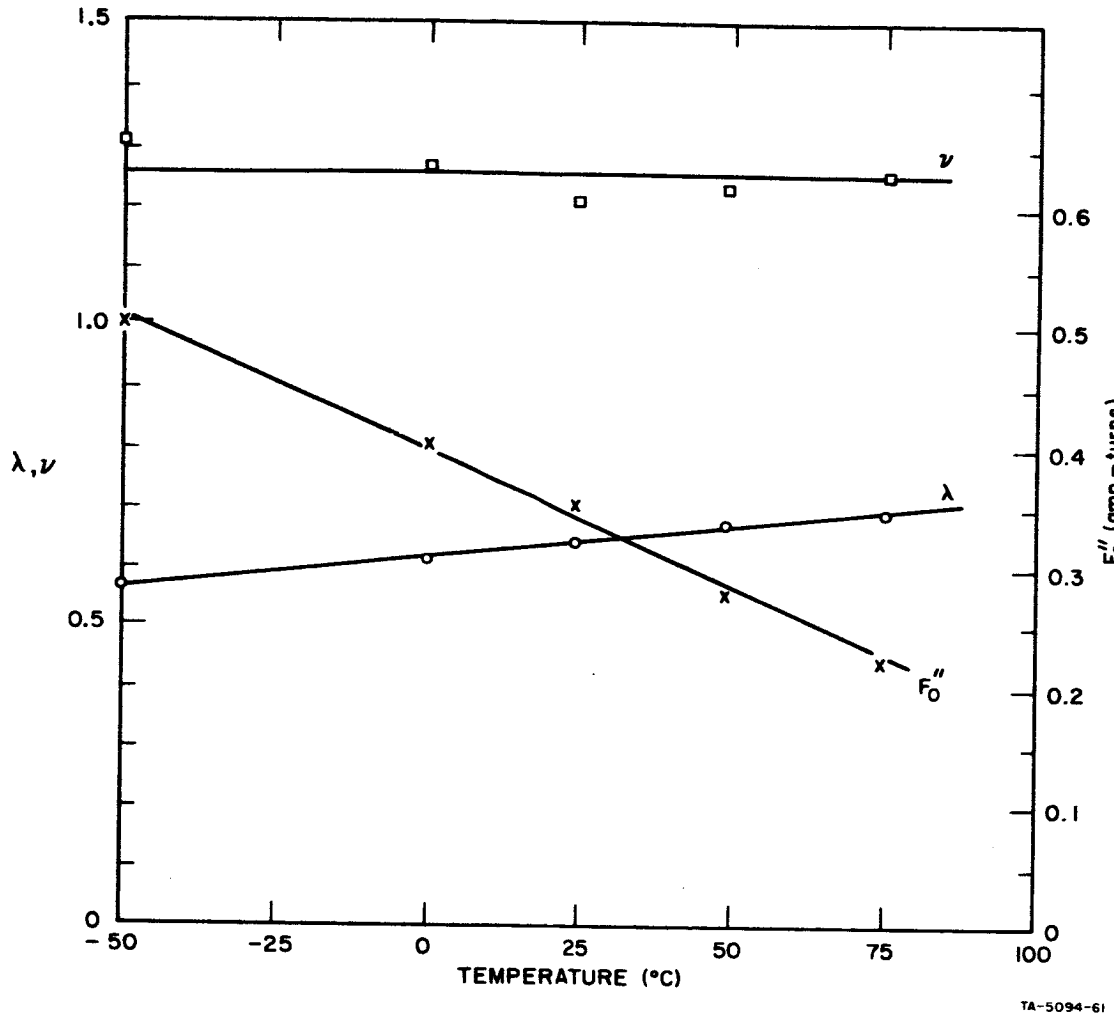


FIG. 49 $\dot{\phi}_p(F)$ PARAMETERS vs. TEMPERATURE FOR CORE K-1

4. RAMP-*F* SWITCHING

The switching properties of a core with ramp-*F* excitation have been discussed in Sec. III-A for room temperature. It was shown that λ needs to be altered (lowered for Core E-6) from its step-*F* value in order that $\dot{\phi}_p(k)$ can be properly described by the parabolic model. It was also shown that the experimental $kt_p(k)$ curve has a somewhat lower slope than the computed curve, so that exact agreement can only be obtained at one k value. We now wish to determine what effect temperature variations have on these properties. Cores I-4 and K-1 were both investigated over the temperature range −50°C to +75°C. The equipment and the experimental technique are as described in Sec. III-A-4. The amplitude and duration of the CLEAR pulses are given in Sec. III-B-2.

The effects of temperature on $\dot{\phi}_p(k)$ is shown in Fig. 50 for Core I-4 and in Fig. 51 for Core K-1. At first glance these curves seem to vary haphazardly, but this is not actually the case. The general trend is for the lowest temperature curve to be on top at high k values, and on the bottom for low k values, so that each curve crosses all the others in going from low-k to high-k values. The only exception is that the curves for Core I-4 are not extended to low-enough k values for this crossing to be completed. Equation (102) for $\dot{\phi}_p(k)$ was derived by using the parabolic model with no static $\phi(F)$ limiting included [*cf.*, Eq. (94) with Eq. (105)]. Equation (102) results in a straight line (see Fig. 30) when $\dot{\phi}_p(k)$ is plotted with log-log scales. The slope of this line was shown (see Fig. 30) to be dependent only upon ν. Thus, if ν is constant with temperature, as it practically is in Core K-1 for step-F switching, parallel lines would result so that no crossing would be obtained. However, if ϕ_d is introduced into the model as shown in Eq. (105), then log $\dot{\phi}_p$(log k) curves downward at low k values. This makes crossing of the curves possible even if ν is constant with temperature. Curves of $\dot{\phi}(k)$ using the ϕ_d parabolic model and step-F parameters were computed for Core K-1. The resulting curves showed

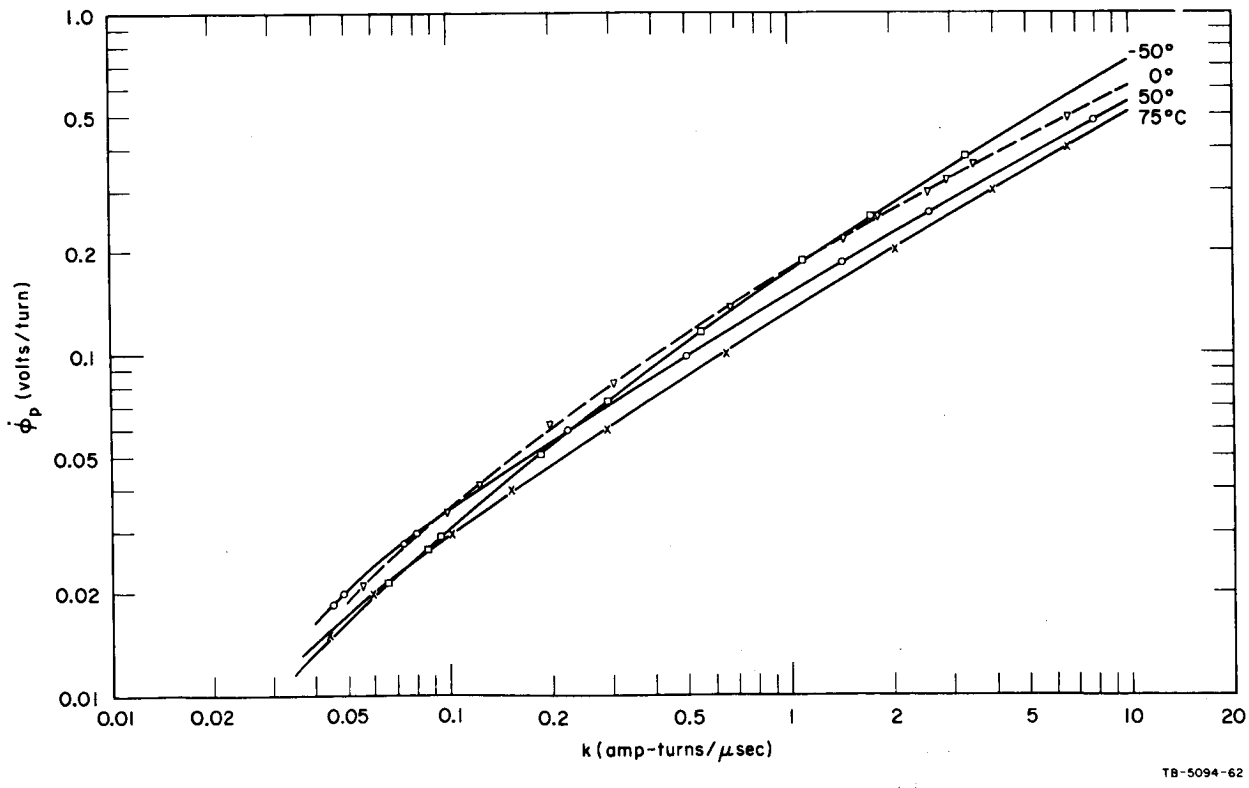


FIG. 50 $\dot{\phi}_p(k)$ CURVES OF CORE I-4 vs. TEMPERATURE WITH RAMP F(t)

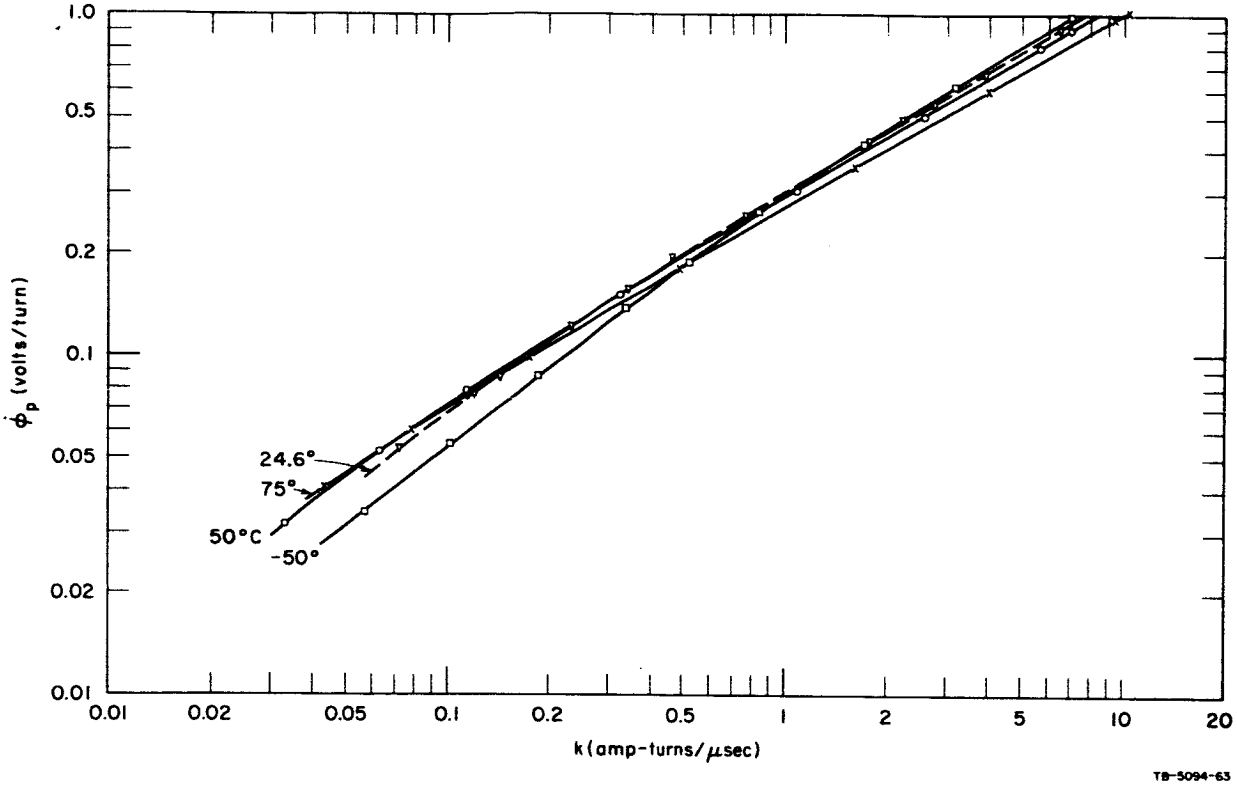


FIG. 51 $\dot{\phi}_p(k)$ CURVES OF CORE K-1 vs. TEMPERATURE WITH RAMP F(t)

no tendency for crossing except for the −50°C curve which crossed all the other three curves (*i.e.*, for 24.6°, +50°, and +75°C). Therefore, most, but not all, of the tendency for the experimental curves to cross is due to the parameters (namely λ and F_0'') varying from their step-F values. It was noted in Sec. III-A-5 that decreasing F_0'' increases the curvature in the computed log-log plot of $\dot{\phi}_p(k)$. Thus, the reduced F_0'' values, F_{0r}'', required for ramp-F switching will increase the tendency, for the $\dot{\phi}_p(k)$ curves to cross. The slight increase in ν with temperature for Core I-4 makes the temperature variations more difficult to analyze.

The effects of temperature upon t_p can be examined from log-log plots of $kt_p(k)$. These curves are shown in Figs. 52 and 53 for Core I-4 and K-1 respectively. The major change with temperature is seen to be a vertical displacement. Examination of Eq. (101) or Eq. (103) shows that this can be caused by variations in λ, F_0'', ν, ϕ_s, and ϕ_r. Only one of these parameters, F_0'' can be altered without much affecting the $\dot{\phi}_p(k)$ curves. Equation (102) for $\dot{\phi}_p$ is independent of F_0''. Thus, for ramp-F switching, as for step-F switching, it is F_0'' which varies most significantly with

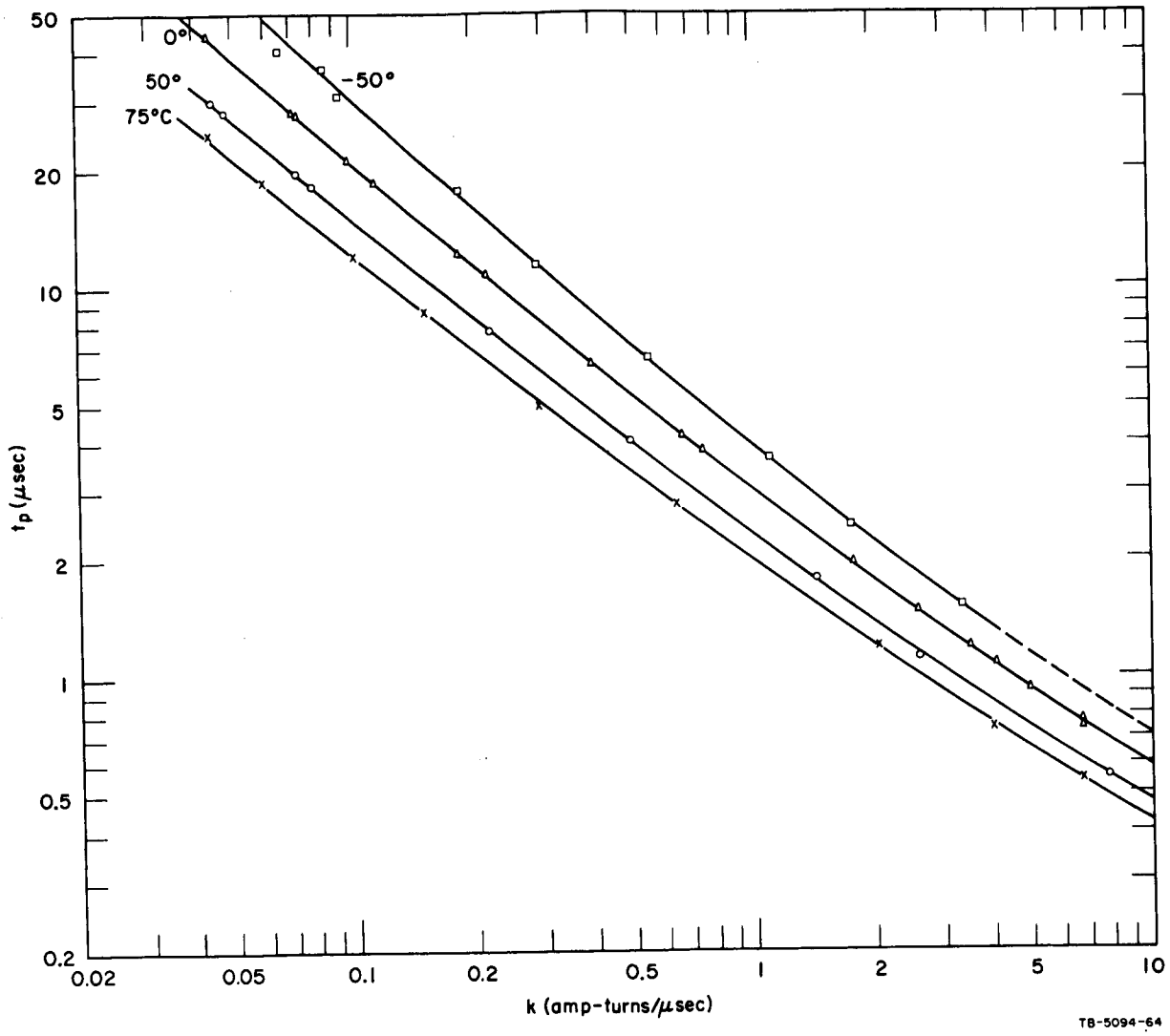


FIG. 52 $t_p(k)$ CURVES OF CORE I-4 vs. TEMPERATURE WITH RAMP F(t)

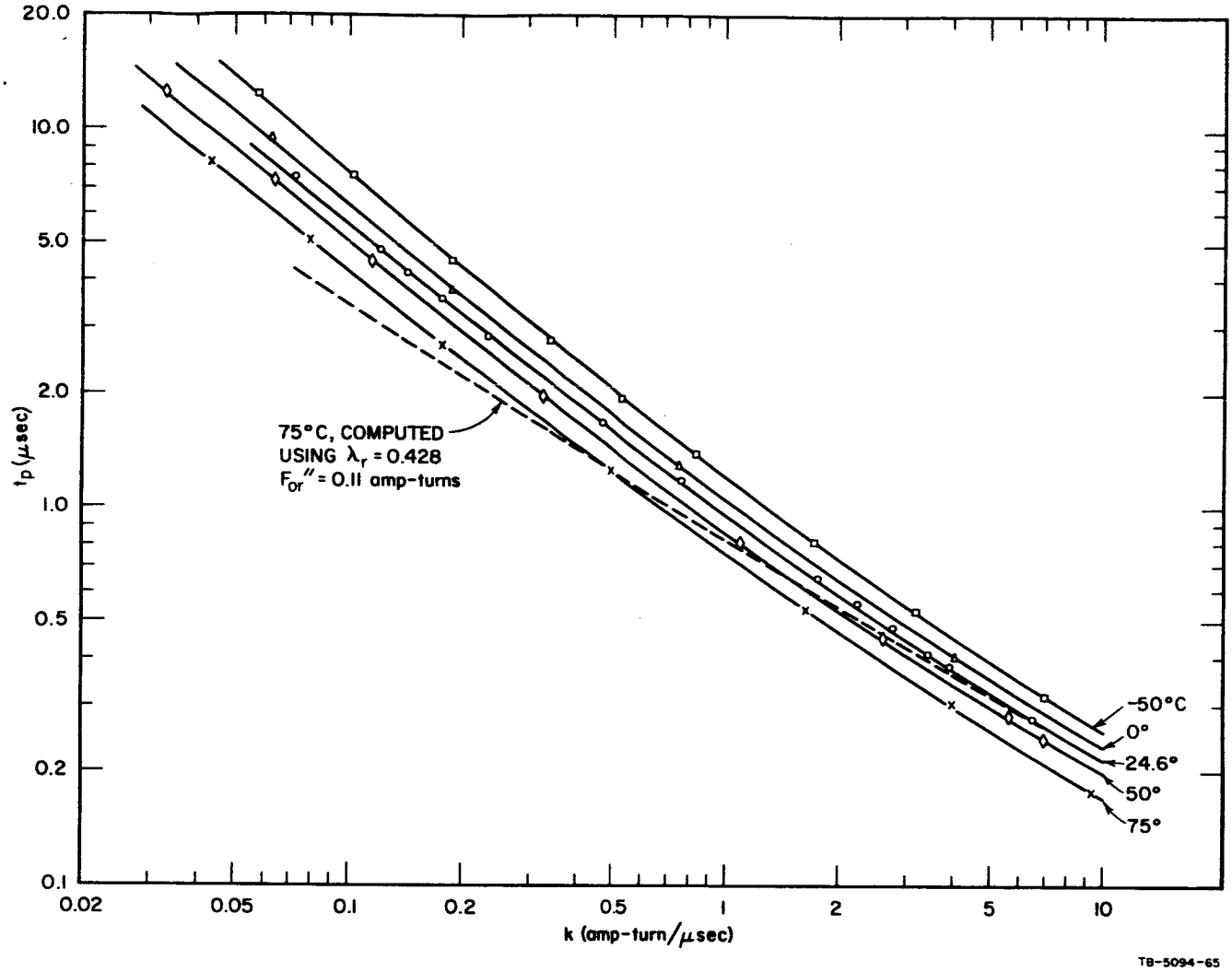


FIG. 53 $t_p(k)$ CURVES OF CORE K-1 vs. TEMPERATURE WITH RAMP F(t)

temperature. This would be expected. However, there is still the problem of getting the experimental and computed $kt_p(k)$ curves to agree for all k values. This was discussed for Core E-6 at room temperature in Sec. III-A-5 (see Fig. 34). To illustrate this problem for Core K-1, a computed curve is included in Fig. 53 for 75°C. It is shown as a dashed line. The value of λ_r for this curve was determined using Eq. (109). The value of F_{0r}'' as determined from Eq. (112) for k = 6 is approximately -0.24 amp-turns which seems unreasonable. Since a negative F_{0r}'' seems unreasonable, a positive value equal to 1/2 F_0'' was used. This resulted in good agreement for $\dot{\phi}_p(k)$. Note, however, that the computed $t_p(k)$ curve in Fig. 53 crosses the corresponding experimental curve at about k = 0.45. The computed curve at k = 6 is considerably above the experimental curve.

Thus F''_{0r} would have to be reduced a lot more to give agreement at $k = 6$. A new computation of F''_{0r} using Eq. (112) and the value of $t_{p\,calc}$ which was computed by using $\lambda_r = 0.428$ and $F''_{0r} = 0.11$ gives the same negative value, -0.24, previously determined. Thus it does seem to be necessary to have a negative value of F''_{0r} in order to have t_p agree at $k = 6$. The 75°C curve for Core K-1 is the only one which resulted in a negative value for F''_{0r}; however, the values at +50°C and 24.6°C were also unreasonably low. This problem has not yet been resolved, but is probably closely related to the general disagreement between the computed and experimental $t_p(k)$ curves. This requires further investigation. Because of the problem concerning $t_p(k)$ and F''_{0r}, no meaningful plot of F''_{0r} *vs.* temperature can be given.

The value of λ_r computed for Core I-4 at 75°C from Eq. (109) was 0.114; only slightly reduced from the step-F value of 0.121. Using this value of λ_r and $F''_{0r} = F''_0 = 0.800$ amp-turns to compute $t_p(k)$ gave very good agreement with the experimental curve for 75°C. Thus Core I-4 behaves much differently than Cores E-6 and K-1 in this respect. The agreement for other temperatures was not as good as at 75°C, but much better than any of the $t_p(k)$ curves for Core K-1.

Values of λ_r were determined for each temperature for both Cores I-4 and K-1. This was done by using Eq. (109) and the values of $\dot{\phi}_{p\,calc}$ and $t_{p\,calc}$ computed from step-F parameters. These values of λ_r were verified by using them to compute $\dot{\phi}_p(k)$ as discussed in Sec. III-A-3. Since F''_{0r} influences $\dot{\phi}_p(k)$ only slightly, the values for F''_0 were used for Core I-4 and the values $1/2\ F''_0$ were used for Core K-1. In all cases the computed $\dot{\phi}_p(k)$ was in good agreement with the experimental curves. These values of λ_r are plotted *vs.* temperature in Fig. 54 for Core I-4 and Fig. 55 for Core K-1. The step-F value of λ is also plotted for comparison. The λ_r for Core K-1 is much lower than λ over the entire temperature range, as was expected. The small number and the scatter of the data points makes it impossible to say for sure whether λ actually varies, as shown by the smooth curve, or whether it is approximately constant with temperature. About all we can assume is that the λ_r temperature coefficient is roughly zero at room temperature.

The λ_r curve for Core I-4 is surprising in two respects. It is not very much lower than λ above 0°C and it is significantly larger than λ at -50°C. If there is any close relationship between the effects of ramp-F

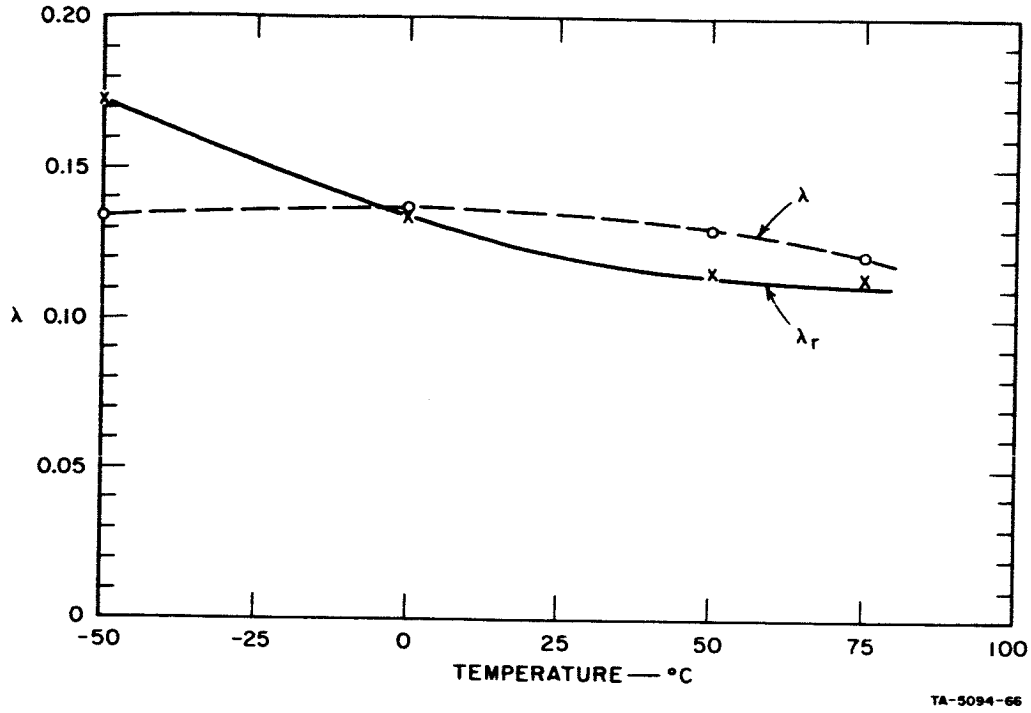


FIG. 54 VARIATIONS OF λ_r AND λ WITH TEMPERATURE FOR CORE I-4

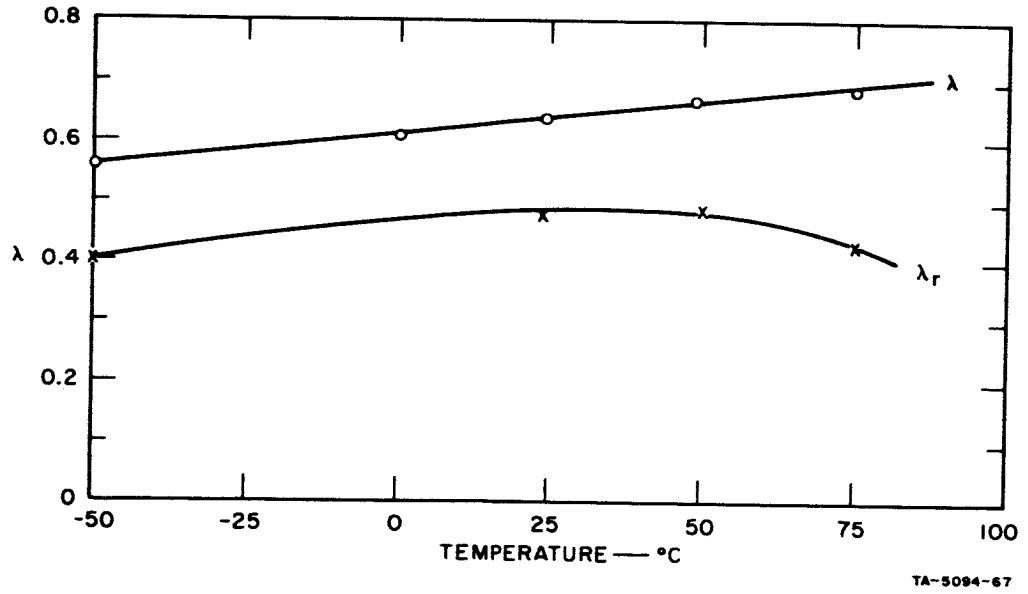


FIG. 55 VARIATIONS OF λ_r AND λ WITH TEMPERATURE FOR CORE K-1

switching and the effects of partial setting, as has been assumed on the basis of other experiments, then the ramp-F switching properties of Core I-4 certainly make this relationship difficult to understand.

It must be noted that the ratio of λ_r to λ is much lower for Cores K-1 and J-1 (discussed in Report 3) than for Cores I-4 and E-6. Cores K-1 and J-1 have two things in common which are not common to Cores I-4 and E-6: They are both Lockheed switch cores (Lockheed No. 100SC1 and 145SC1 respectively) and they both have a relatively large *OD/ID* ratio compared to Cores I-4 and E-6. It cannot yet be stated whether either of these is related to the lower λ_r/λ ratio for these cores than for Cores I-4 and E-6.

5. SUMMARY

The effects of temperature variations on the switching properties were investigated experimentally over the range −50°C to +75°C. This was done for static $\phi(F)$ curves, $\dot{\phi}_p(F)$ curves for step-F switching, and $\dot{\phi}_p(k)$ and $t_p(k)$ curves for ramp-F switching. The static $\phi(F)$ curves were taken for both hard remanent and partially set states.

The core parameters were determined from these data and plotted *vs.* temperature. The parameters H_q, H_n, and F_0'' decreased very markedly with temperature, as expected. The value of ϕ_r and ϕ_s decreased significantly but not as much as H_q and H_n. The values of λ and ν were not very much altered by varying the temperature. Temperature coefficients are given in Table II. Three families of experimental static $\phi(F)$ curves with temperature as a parameter were given: (1) starting from $-\phi_r$; (2) starting from $\phi = 0$ with a positive F; and (3) starting from $\phi = 0$ with a negative F. The core was partially set with a 1-μsec retangular current pulse. Families of $\dot{\phi}_p(F)$, $\dot{\phi}_p(k)$ and $t_p(k)$ were also included.

C. CONCLUSIONS

The flux-switching properties of square-loop ferrite cores, in response to a ramp-F drive, have been investigated as a function of the slope, k, of the ramp. Three cores were tested in the experiments. It was found that the static ϕ_d parabolic model could adequately describe the switching over a limited range of k, provided that λ and F_0'' are given new values λ_r and F_{0r}''. Generally these values need to be lower than λ and F_0'' for step-F switching, but Core I-4 is an exception. There is no need to alter the step-F value of ν. A comparison between experiment

and computation for thin-ring Core E-6 showed that the shape of $\dot{\phi}(t)$ computed by the model is accurate if appropriate values of λ_r and F''_{0r} are used. The experimental and computed $\dot{\phi}_p(k)$ curves could be made to agree over the entire range of k studied (*i.e.*, k = 0.1 to 10 amp-turns/μsec) by using an appropriate value of λ. The computed $\dot{\phi}_p(k)$ curve depends upon F''_{0r} only as a second-order effect. This second-order dependence is a result of including $\phi_d(F)$ in the model.

The computed $t_p(k)$ curves exhibited a tendency to cross the experimental curves. This crossing point can be fixed at any k value by appropriate adjustment of F''_{0r}. Thus, in a practical application, F''_{0r} should be chosen to make $t_p(k)$ agree with experiment near the center of the range of k values involved.

The determination of correct values for λ_r and F''_{0r} for a new core requires that experimental measurements be made with a ramp-F drive. However, only a few experimental points need to be taken.

The effects of temperature upon the switching properties have been experimentally determined for two cores, thin-ring Core I-4 and Core K-1 (see Table II for information on these cores). The temperature range −50°C to +75°C was investigated. In general, the properties changed with temperature in only a quantitative way. The experiments indicate that the parabolic model should be as useful at any temperature in this range as it is at room temperature. The values of the parameters H_q, H_n, ϕ_r, ϕ_s, F''_0, λ, ν, and λ_r have been plotted *vs.* temperature. The most significant effect of increasing temperature is the decrease in the static $\phi(F)$ parameters H_q and H_n, and in the threshold, F''_0, of the $\dot{\phi}_p(F)$ curve. The values of ϕ_r and ϕ_s also decrease with temperature but less severely than H_q, H_n, and F''_0. The values of λ and ν are not strongly dependent upon temperature. The small variation in λ with increasing temperature for Core I-4 was, if anything, downward. Core K-1, on the other hand, exhibited a measurable increase in λ with an increase in temperature (about 0.16 percent per °C at 30°C).

The effects of temperature upon the static $\phi(F)$ curves starting from a partially set state are difficult to describe quantitatively, since we do not yet have equations to describe these curves. The general shape, relative to the major static $\phi(F)$ curves, was largely unchanged with temperature (see Figs. 44 and 45).

The $\dot{\phi}_p(k)$ curves for ramp-F switching were not very strongly influenced by changes in temperature. Presumably this is a result of $\dot{\phi}_p(k)$ being nearly independent of F_0'' which is the major temperature-dependent parameter.

The $t_p(k)$ curves are shifted downward by increasing temperature. The shape is relatively unchanged in other aspects. This downward shift is probably a result of the decrease in F_0'' with temperature, although the tendency for the computed $t_p(k)$ curves to cross the experimental curves is not yet understood. An additional problem with $t_p(k)$ showed up at +50° and +75°C. This is the apparent negative, or very low positive, value for F_{0r}''. The fact that the OD/ID ratio of the core has not been accounted for in the model [except for $\phi_d(F)$] may be responsible for some of this trouble. These problems with $t_p(k)$ and F_0'' require further investigation.

The ramp-F switching properties of Core I-4, in relation to step-F switching properties, are different than for the other three cores which have been tested (this includes Cores E-6, K-1 and a one-point check on Core J-1 in Report 3). The relationship of parameter variations for ramp-F switching and switching from a partially set state is not yet understood; although, a close relationship must exist. Except for Core I-4, λ and F_0'' are reduced from step-F values for both cases. Core I-4, on the other hand, exhibited a large reduction in λ for a partially set state (see p. 113 of Report 3) but only a small reduction for ramp-F switching (even an increase at -50°C). This relationship appears to require rather extensive work before it can be clearly understood. However, the presently used switching model can be used in many practical applications where $F(t)$ is approximately a ramp function.

APPENDIX A

COMPUTER PROGRAM FOR PROCEDURE $\dot{\phi}_\epsilon(\dot{F}, \Delta t, NV, \dot{\phi}'_\epsilon)$

Language: ALGOL 60.

Program Description: Computes $\dot{\phi}_\epsilon$ and $\dot{\phi}'_\epsilon$ for given values of $\dot{F}$, Δt, and NV (negligible value of $\dot{\phi}_\epsilon$ below which $\dot{\phi}_\epsilon$ is assumed to be zero). Switching parameter ϵ is global.

Identifiers:*

Identifier	Symbol	Identifier	Symbol
DELT	Δt	PHIDOTE	$\dot{\phi}_\epsilon$
EPS	ϵ	PHIDOTEPRIME	$\dot{\phi}'_\epsilon$
FDOT	$\dot{F}$	PHIDTE	$\dot{\phi}_\epsilon$
NV	NV		

Program:

```
REAL PROCEDURE PHIDOTE (FDOT, DELT, NV, PHIDOTEPRIME);                    PHDOTE01
COMMENT: THIS PROCEDURE COMPUTES PHIDOTE AND PHIDOTEPRIME VS. FDOT        PHDOTE02
AND DELT. PHIDOTE IS ASSUMED ZERO IF BELOW NV. THE PROCEDURE USES         PHDOTE03
GLOBAL PARAMETER EPS.;                                                    PHDOTE04
VALUE FDOT, DELT, NV;                                                     PHDOTE05
REAL FDOT, DELT, NV, PHIDOTEPRIME;                                        PHDOTE06
BEGIN REAL PHIDTE;                                                        PHDOTE07
   PHIDTE ← EPS × FDOT;                                                   PHDOTE08
   PHIDOTE ← IF PHIDTE > NV THEN PHIDTE ELSE 0;                           PHDOTE09
   PHIDOTEPRIME ← IF PHIDTE > NV THEN EPS / DELT ELSE 0;                  PHDOTE10
END OF PHIDOTE;                                                           PHDOTE11
```

* Listed in alphabetic order.

APPENDIX B

COMPUTER PROGRAM FOR PROCEDURE $\dot{\phi}_i(F, t, T_i, \dot{\phi}_i')$

Language: ALGOL 60

Program Description: Computes $\dot{\phi}_i$, T_i, and $\dot{\phi}_i'$ for given values of F and t. Switching parameters λ_i, ν_i, C_i and F_i are global.

Identifiers:

Identifier	Symbol	Identifier	Symbol
CI	C_i	PHIDOTI	$\dot{\phi}_i$
F	F	PHIDOTIPRIME	$\dot{\phi}_i'$
FDEX	F_{ex}	PHIDTI	$\dot{\phi}_i$
FI	F_i	T	t
LAMBDAI	λ_i	TI	T_i
NUI	ν_i		

Program:

```
REAL PROCEDURE PHIDOTI (F, T, TI, PHIDOTIPRIME);                    PHDOTI01
COMMENT: THIS PROCEDURE COMPUTES PHIDOTI AND PHIDOTIPRIME VS. F, T  PHDOTI02
, AND TI. THE PROCEDURE USES GLOBAL PARAMETERS LAMBDAI, NUI, CI,    PHDOTI03
AND FI.;                                                            PHDOTI04
VALUE F, T;                                                         PHDOTI05
REAL F, T, TI, PHIDOTIPRIME;                                        PHDOTI06
BEGIN REAL FDEX, PHIDTI;                                            PHDOTI07
   FDEX ← F - FI;                                                   PHDOTI08
   IF TI ≠ 0 THEN PHIDTI ← LAMBDAI × FDEX ↑ NUI × EXP ((TI - T) ×   PHDOTI09
   FDEX / CI) ELSE                                                  PHDOTI10
   BEGIN PHIDTI ← 0;                                                PHDOTI11
      IF FDEX > 0 THEN                                              PHDOTI12
        BEGIN TI ← T; PHIDTI ← LAMBDAI × FDEX ↑ NUI × EXP ((TI - T) PHDOTI13
        × FDEX / CI) ;                                              PHDOTI14
        END;                                                        PHDOTI15
   END;                                                             PHDOTI16
   PHIDOTI ← IF PHIDTI > 0.0010 × LAMBDAI × FDEX THEN PHIDTI ELSE   PHDOTI17
   0;                                                               PHDOTI18
   PHIDOTIPRIME ← IF PHIDTI > 0.0010 × LAMBDAI × FDEX THEN PHIDTI   PH4OTI19
   × (NUI / FDEX - (T - TI) / CI) ELSE 0;                           PHDOTI20
END PHIDOTI;                                                        PHDOTI21
```

APPENDIX C

COMPUTER PROGRAM FOR PROCEDURE $\dot{\phi}_{ma}(F,\phi,\phi_d,\dot{\phi}'_{ma})$

Language: ALGOL 60

Program Description: Computes $\dot{\phi}_{ma}$, ϕ_d, and $\dot{\phi}'_{ma}$ for given values of F and ϕ. Switching parameters l_i, l_o, ϕ_r, ϕ_s, H_a, H_q, H_n, λ, F''_0, ν, ρ_p, F_0, F_B, F_{12}, F_{23}, V_1 and V_2 are global.

Identifiers:

Identifier	Symbol	Identifier	Symbol
F	F	PHI	ϕ
FB	F_B	PHID	ϕ_d
F0	F_0	PHIDOTMA	$\dot{\phi}_{ma}$
F0PP	F''_0	PHIDOTMAPRIME	$\dot{\phi}'_{ma}$
F12	F_{12}	PHIDOTP	$\dot{\phi}_p$
F23	F_{23}	PHIDOTPPRIME	$\dot{\phi}'_p$
HA	H_a	PHIDPRIME	ϕ'_d
HN	H_n	PHIR	ϕ_r
HQ	H_q	PHIS	ϕ_s
LAMBDA	λ	ROP	ρ_p
LI	l_i	V1	V_1
LO	l_o	V2	V_2
NU	ν		

Program:

```
REAL PROCEDURE PHIDOTMA (F, PHI, PHID, PHIDOTMAPRIME);                    PHDTMA01
COMMENT: THIS PROCEDURE COMPUTES PHIDOTMA, PHID, AND PHIDOTMAPRIME        PHDTMA02
FOR GIVEN VALUES OF F AND PHI. GLOBAL CORE PARAMETERS ARE: LI,LO,         PHDTMA03
PHIR,PHIS,HA,HQ,HN,LAMBDA,F0PP,NU,ROP,F0,FB,F12,F23,V1,AND V2.;           PHDTMA04
REAL F, PHI, PHID, PHIDOTMAPRIME;                                         PHDTMA05
BEGIN REAL PHIDPRIME, PHIDOTP, PHIDOTPPRIME;                              PHDTMA06
   COMMENT: COMPUTE PHID AND PHIDPRIME VS. F.;                            PHDTMA07
   IF F ≤ F12 THEN                                                        PHDTMA08
   BEGIN PHID ← V1 × F × LN ((F - HA × LO) / (F - HA × LI)) - PHIR        PHDTMA09
      ;                                                                   PHDTMA10
      PHIDPRIME ← V1 × (LN ((F - HA × LO) / (F - HA × LI)) + F × (        PHDTMA11
      1 / (F - HA × LO) - 1 / (F - HA × LI)))                             PHDTMA12
   END;                                                                   PHDTMA13
```

```
                                                                    PHDTMA14
  IF F12 < F AND F ≤ F23 THEN                                       PHDTMA15
  BEGIN PHID ← V2 × (F / HQ - LI + F × (1 / HN - 1 / HQ) × LN ((1   PHDTMA16
     - HN / HQ) / (1 - HN × LI / F))) - PHIR;                       PHDTMA17
     PHIDPRIME ← V2 × (1 / HQ + (1 / HN - 1 / HQ) × (LN (F × (1 -   PHDTMA18
     HN / HQ) / (F - HN × LI)) - HN × LI / (F - HN × LI)))          PHDTMA19
  END;                                                              PHDTMA20
  IF F23 < F THEN                                                   PHDTMA21
  BEGIN PHID ← V2 × (LO - LI + F × (1 / HN - 1 / HQ) × LN ((F -     PHDTMA22
     HN × LO) / (F - HN × LI))) - PHIR;                             PHDTMA23
     PHIDPRIME ← V2 × (1 / HN - 1 / HQ) × (LN ((F - HN × LO) / (F   PHDTMA24
     - HN × LI)) + F × HN × (LO - LI) / ((F - HN × LO) × (F - HN    PHDTMA25
     × LI)))                                                        PHDTMA26
  END;                                                              PHDTMA27
  COMMENT: COMPUTE PHIDOTP AND PHIDOTPPRIME VS. F.;                 PHDTMA28
  IF F ≤ FOPP THEN                                                  PHDTMA29
  BEGIN PHIDOTP ← 0;                                                PHDTMA30
     PHIDOTPPRIME ← 0                                               PHDTMA31
  END;                                                              PHDTMA32
  IF FOPP < F AND F ≤ FB THEN                                       PHDTMA33
  BEGIN PHIDOTP ← LAMBDA × (F - FOPP) * NU;                         PHDTMA34
     PHIDOTPPRIME ← LAMBDA × NU × (F - FOPP) * (NU - 1)             PHDTMA35
  END;                                                              PHDTMA36
  IF FB < F THEN                                                    PHDTMA37
  BEGIN PHIDOTP ← ROP × (F - F0);                                   PHDTMA38
     PHIDOTPPRIME ← ROP                                             PHDTMA39
  END;                                                              PHDTMA40
  COMMENT: COMPUTE PHIDOTMA AND PHIDOTMAPRIME.;                     PHDTMA41
  PHIDOTMA ← IF PHID - PHI > 0.001 × PHIR THEN PHIDOTP × (1 - ((2   PHDTMA42
  × PHI + PHIR - PHID) / (PHIR + PHID)) * 2) ELSE 0;                PHDTMA43
  PHIDOTMAPRIME ← IF PHID - PHI > 0.0010×PHIR THEN (1 - ((2 × PHI   PHDTMA44
  + PHIR - PHID) / (PHIR + PHID)) * 2) × PHIDOTPPRIME + 4 × PHIDOTP PHDTMA45
  × (2 × PHI + PHIR - PHID) × (PHI + PHIR) × PHIDPRIME / (PHIR +    PHDTMA46
  PHID) * 3 ELSE 0 ;                                                PHDTMA47
END PHIDOTMA;
```

APPENDIX D

COMPUTER PROGRAMS FOR $\dot{\phi} \cong \dot{\phi}_i + \dot{\phi}_{ma}$

1. PROCEDURE $\dot{\phi}(F, \phi, \phi_d, \dot{\phi}')$

Language: ALGOL 60

Program Description: Computes $\dot{\phi}$, ϕ_d, and $\dot{\phi}'$ for given values of F and ϕ. Switching parameters l_i, l_o, ϕ_r, ϕ_s, H_a, H_q, H_n, λ, F_0'', ν, ρ_p, F_0, and F_B are global.

Identifiers:

(1) *Analytical identifiers*

Same as in Appendix C, except for PHIDOT ($\dot{\phi}$) and PHIDOTPRIME ($\dot{\phi}'$).

(2) *Auxiliary identifiers*

Identifier	Description
DONE	Boolean variable
OK	Label

Program:

```
REAL PROCEDURE PHIDOT(F,PHI,PHID,PHIDOTPRIME);                          PHIDOT01
    COMMENT: THIS PROCEDURE COMPUTES PHIDOT, PHID, AND PHIDOTPRIME      PHIDOT02
        FOR GIVEN VALUES OF F AND PHI.  CORE PARAMETERS, WHICH MUST BE  PHIDOT03
        SUPPLIED FROM OUTSIDE THE PROCEDURE, ARE:                       PHIDOT04
        LI, LO, PHIR, PHIS, HA, HQ, HN, LAMBDA, FOPP, NU, ROP, F0, FB ; PHIDOT05
    REAL F, PHI, PHID, PHIDOTPRIME;                                     PHIDOT06
    BEGIN                                                               PHIDOT07
    REAL PHIDPRIME, PHIDOTP, PHIDOTPPRIME ;                             PHIDOT08
    OWN REAL F12, F23, V1, V2 ;                                         PHIDOT09
    LABEL OK;                                                           PHIDOT10
    OWN BOOLEAN DONE;                                                   PHIDOT11
    COMMENT: COMPUTE F12, F23, V1, AND V2 ONLY ONCE.;                   PHIDOT12
        IF DONE THEN GO TO OK;                                          PHIDOT13
            DONE ← TRUE;                                                PHIDOT14
            F12 ← HQ×LI ;                                               PHIDOT15
            F23 ← HQ×LO ;                                               PHIDOT16
            V1 ← (PHIS-PHIR)/((LO-LI)×HA);                              PHIDOT17
            V2 ← (PHIS+PHIR)×HQ/((LO-LI)×HN);                           PHIDOT18
    OK:                                                                 PHIDOT19
```

Lines PHDTMA07 through PHDTMA39
of
PROCEDURE $\dot{\phi}_{ma}(F,\phi,\phi_d,\dot{\phi}')$, APPENDIX C

```
COMMENT: COMPUTE PHIDOT AND PHIDOTPRIME.;                              PHIDOT57
PHIDOT ← IF PHID-PHI > 0.001×PHIR THEN PHIDOTP×(1-((2×PHI+PHIS         PHIDOT58
-PHID)/(PHIS+PHID))*2) ELSE 0 ;                                        PHIDOT59
PHIDOTPRIME ← (1-((2×PHI+PHIS-PHID)/(PHIS+PHID))*2)×PHIDOTPPRIME       PHIDOT60
+4×PHIDOTP×(2×PHI+PHIS-PHID)×(PHI+PHIS)×PHIDPRIME/(PHIS+PHID)*3        PHIDOT61
END PHIDOT;                                                            PHIDOT62
```

2. PROCEDURE $\dot{\phi}(F,\ \phi,\ \dot{\phi}',\ \dot{\phi}^*)$

Language: ALGOL 60

Program Description: Computes $\dot{\phi}$, $\dot{\phi}'$, and $\dot{\phi}^*$ for given values of F and ϕ. Switching parameters l_i, l_o, ϕ_r, ϕ_s, H_a, H_q, H_n, λ, F_0'', ν, ρ_p, F_0, F_B, F_{12}, F_{23}, V_1, and V_2 are global.

Identifiers:

Same as in Appendix C, except for PHIDOT ($\dot{\phi}$) and PHIDOTPRIME ($\dot{\phi}'$).

Program:

```
REAL PROCEDURE PHIDOT (F, PHI, PHIDOTPRIME, PHIDOTSTAR);               PHIDOT01
COMMENT: THIS PROCEDURE COMPUTES THE MAIN PHIDOT, PHIDOTPRIME AND      PHIDOT02
PHIDOTSTR FOR GIVEN VALUES OF F AND PHI. THE PROCEDURE USES THE        PHIDOT03
FOLLOWING GLOBAL PARAMETERS: LI, LO, PHIR, PHIS, HA, HQ, HN, LAMBDA,   PHIDOT04
FOPP, NU, ROP, F0, FB, F12, F23, V1,AND V2;                            PHIDOT05
                                                                       PHIDOT06
VALUE F, PHI;                                                          PHIDOT07
REAL F, PHI, PHIDOTPRIME, PHIDOTSTAR;                                  PHIDOT08
BEGIN REAL PHIDPRIME, PHID, PHIDOTP, PHIDOTPPRIME;
```

Lines PHDTMA07 through PHDTMA39
of
PROCEDURE $\dot{\phi}_{ma}(F,\phi,\phi_d,\dot{\phi}')$, APPENDIX C

```
    COMMENT: COMPUTE PHIDOT, PHIDOTPRIME AND PHIDOTSTAR.;              PHIDOT41
    PHIDOT ← IF PHID - PHI > 0.001 × PHIR THEN PHIDOTP × (1 - ((2 ×    PHIDOT42
    PHI + PHIS - PHID) / (PHIS + PHID)) * 2) ELSE 0;                   PHIDOT43
    PHIDOTPRIME ← IF PHID - PHI > 0.0010 × PHIR THEN (1 - ((2 × PHI +  PHIDOT44
    PHIS - PHID) / (PHIS + PHID)) * 2) × PHIDOTPPRIME + 4 × PHIDOTP × ( PHIDOT45
    2 × PHI + PHIS - PHID) × (PHI + PHIS) × PHIDPRIME / (PHIS + PHID)  PHIDOT46
                                                                       PHIDOT47
    * 3 ELSE 0;                                                        PHIDOT48
    PHIDOTSTAR ← - 4 × PHIDOTP × (2 × PHI + PHIS - PHID) / (PHIS +     PHIDOT49
    PHID) * 2;                                                         PHIDOT50
END PHIDOT;
```

APPENDIX E

COMPUTER PROGRAM FOR $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$ AND $\dot{\phi}$ OF UNLOADED CORE

Language: ALGOL 60

Program Description: Computes i_D, $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$, $\dot{\phi}$, ϕ, ϕ_d and F versus t for given core parameters and drive function.

Identifiers:

(1) *Analytical identifiers*

Identifier	Symbol	Identifier	Symbol
CAPID	I_D	ND	N_D
CI	C_i	NU	ν
DELT	Δt	NUI	ν_i
EPS	ϵ	PHIC	ϕ
F	F	PHIC1	ϕ_{n-1}
FB	F_B	PHIC2	ϕ_{n-2}
FDOTC	$\dot{F}$	PHIDC	ϕ_d
FI	F_i	PHIDOTC	$\dot{\phi}$
F0	F_0	PHIDOTC1	$\dot{\phi}_{n-1}$
F0I	F_{0i}	PHIDOTEC	$\dot{\phi}_\epsilon$
F0PP	F_0''	PHIDOTEPR	$\dot{\phi}_\epsilon'$
F12	F_{12}	PHIDOTIC	$\dot{\phi}_i$
F23	F_{23}	PHIDOTIPR	$\dot{\phi}_i'$
HA	H_a	PHIDOTMAC	$\dot{\phi}_{ma}$
HQ	H_q	PHIDOTMAPR	$\dot{\phi}_{ma}'$
HN	H_n	PHIR	ϕ_r
ID	i_D	PHIS	ϕ_s
LAMBDA	λ	ROP	ρ_p
LAMBDAI	λ_i	SP	S_p
LI	l_i	T	t
LO	l_o	TAUS	τ_s

Identifier	Symbol	Identifier	Symbol
TF	T_f	TS	T_s
TH	T_h	U	u
TIC	T_i	V1	V_1
TM	t_m	V2	V_2
TR	T_r		

(2) *Auxiliary identifiers*

Identifier	Description
A1	Abbreviation for $\left(S_p - \frac{I_D}{t_m}\right)/(t_m)^2$
A2	Abbreviation for $\left(\frac{3}{2}\frac{I_D}{t_m} - S_p\right)/t_m$
CIRCUITPARAMETERH	Format for the list CIRCUITPARAMETERL.
CIRCUITPARAMETERL	List of drive parameters (I_D, N_D, T_r, u, t_m, and S_p).
CIRCUITPARAMETERS	List of drive parameters (I_D, t_m, S_p, Δt, and plot scales for t, F, ϕ, and type) for input-data cards.
CORENAME	Name of core.
COREPARAMETERH	Format for the list COREPARAMETERL.
COREPARAMETERL	List of core parameters.
COREPARAMETERS	List of core parameters for input-data cards.
COUNT	Index number of Δt.
CTS	Index number of iteration.
C0	$\phi_{n(j=1)} - \phi_{n(j=0)}$.
DELPHIC	$\phi_{n(j)} - \phi_{n(j-1)}$.
FSCALE	F scale (for plotting).
GUESS	Label of location where prediction is made.
K	Index number of Δt for plotting.
LOOP	Label of location where iterative computation begins.
NV	Negligible value of $\dot{\phi}_\epsilon$.
OUTPUTFORMAT	Format for the list OUTPUTVARIABLES.
OUTPUTHEADING	Format for output-column heading.
OUTPUTVARIABLES	List of results (t, i_D, $\dot{\phi}_\epsilon$, $\dot{\phi}_i$, $\dot{\phi}_{ma}$, $\dot{\phi}$, ϕ, ϕ_d, F, and j_{max}).

Identifier	Description
P	Boolean variable determining type of output (for plotting).
PLOTE	Label (for plotting).
PSCALE	$\dot{\phi}$ scale (for plotting).
QUIT	Label of location where computation terminates.
QUITCKT	Label of location where computation of given circuit parameters terminates.
STARTCKT	Label of location where computation for given drive begins.
STARTCORE	Label of location where computation for given core begins.
SWITCHING	Label of location where computation starts for each Δt.
THETA5	$[\phi_{n(j=5)} - \phi_{n(j=4)}]/[\phi_{n(j=1)} - \phi_{n(j=0)}]$
TSCALE	Time scale (for plotting).

Program:

```
ELASTIC AND INELASTIC INITIAL PHIDOT SPIKES OF AN UNLOADED CORE.                001
BEGIN                                                                           002
   COMMENT: DECLARATION OF CORE PARAMETERS.;                                    003
   ALPHA CORENAME;                                                              004
   REAL LI, LO, PHIR, PHIS, HA, HQ, HN, LAMBDA, FOPP, NU, ROP, F0, FB           005
   , V1, V2, F12, F23, EPS, LAMBDAI, NUI, CI, F0I, FI;                          006
   COMMENT: DECLARATION OF CIRCUIT PARAMETERS.;                                 007
   REAL CAPID, ND, TR, U, TM, SP, TS, TAUS, TH, TF, A1, A2;                     008
   COMMENT: DECLARATION OF VARIABLES.;                                          009
   REAL T, DELT, TIC, ID, F, FDOTC, PHIDOTC, PHIDOTC1, PHIDOTMAC,               010
   PHIDOTEC, PHIDOTIC, PHIDOTMAPR, PHIDOTEPR, PHIDOTIPR, PHIC, PHIC1,           011
   PHIC2, DELPHIC, C0, THETA5, PHIDC;                                           012
   COMMENT: DECLARATION OF MISCELLANEOUS.;                                      013
   REAL NV, TSCALE, FSCALE, PSCALE ;                                            014
   INTEGER LINES, COUNT, CTS, K;                                                015
   BOOLEAN P ;                                                                  016
   LABEL STARTCORE, STARTCKT, SWITCHING, GUESS,LOOP,QUITCKT,PLOTE,QUIT;         017
   REAL ARRAY PHIDTA, QDTA, TIMEA [0 : 500];                                    018
   COMMENT: DECLARATION OF FILE, LISTS, AND FORMATS.;                           019
   FILE CR 0 (2, 10);                                                           020
   FILE F1 1 (2, 15);                                                           021
   LIST COREPARAMETERS (CORENAME, LI, LO, PHIR, PHIS, HA, HQ, HN,               022
   LAMBDA, FOPP, NU, ROP, F0, FB, LAMBDAI, NUI, CI, F0I);                       023
   LIST COREPARAMETERL (CORENAME, LI × 103, LO × 103, PHIR × 108,               024
   PHIS × 108, HA, HQ, HN, LAMBDA, FOPP, NU, ROP, F0, FB, EPS × 109,            025
   LAMBDAI, NUI, CI × 106, F0I);                                                026
   LIST CIRCUITPARAMETERS(CAPID,TM,SP,DELT,TSCALE,FSCALE,PSCALE,P);             027
   LIST CIRCUITPARAMETERL (CAPID, ND, TR × 106, U × 10-6, TM × 106,             028
   SP × 10-6);                                                                  029
```

```
    LIST OUTPUTVARIABLES (T x 1Q6, ID, PHIDOTEC, PHIDOTIC, PHIDOTMAC,          030
    PHIDOTC, PHIC x 1Q8, PHIDC x 1Q8, F, CTS);                                 031
    FORMAT COREPARAMETERH ("CORE ", A6, X4, "LI=", F8.3, X4, "LO=",            032
    F8.3, X4, "PHIR=", F8.3, X4, "PHIS=", F8.3, X4, "HA=", F8.3, X4, "HQ       033
=", F8.3, X4, "HN=", F8.3 / X15, "LAMBDA=", F8.5, X4, "FOPP=", F8.3,           034
    X4, "NU=", F8.3, X4, "ROP=", F8.3, X4, "F0=", F8.3, X4, "FB=",             035
    F8.3 / X15, "EPS=", F8.4, "Q-9", X4, "LAMBDAI=", F8.5, X4, "NUI=",         036
    F8.3, X4, "CI=", F8.3, "Q-6", X4, "F0I= ", F8.3 /);                        037
    FORMAT CIRCUITPARAMETERH ("DRIVE PARAMETERS", X8, "CAPID=", F8.4,           038
    X4, "ND=", I2, X4, "TR=", F8.3, X4, "U=", F6.2, "Q6", X4, "TM=",           039
    F8.3, X4, "SP=", F6.2, "Q6" /);                                            040
    FORMAT OUTPUTHEADING (/ X6, "T", X11, "ID", X5, "PHIDOTEC", X4, "PHI       041
DOTIC", X4, "PHIDOTMAC", X4, "PHIDOTC", X8, "PHIC", X7, "PHIDC", X9,          042
    "F", X9, "CTS" /);                                                         043
    FORMAT OUTPUTFORMAT (F9.3, F12.3, 4 F12.5, 2 F12.2, F12.3, I10);           044
    COMMENT: DECLARATION OF PROCEDURES.;                                       045
```

PROCEDURE $\dot{\phi}_{\epsilon}(\dot{F},\Delta t,NV,\dot{\phi}'_{\epsilon})$, APPENDIX A
PROCEDURE $\dot{\phi}_{i}(F,t,T_{i},\dot{\phi}'_{i})$, APPENDIX B
PROCEDURE $\dot{\phi}_{ma}(F,\phi,\phi_{d},\dot{\phi}'_{ma})$. APPENDIX C

```
    REAL PROCEDURE TANH (X1);                                                  TANH0001
    VALUE X1;                                                                  TANH0002
    REAL X1;                                                                   TANH0003
    BEGIN REAL Y;                                                              TANH0004
       Y ← EXP (X1 + X1);                                                      TANH0005
       TANH ← (Y - 1.0) / (Y + 1.0)                                            TANH0006
    END TANH;                                                                  TANH0007

    REAL PROCEDURE SECH (X);                                                   SECH0001
    VALUE X;                                                                   SECH0002
    REAL X;                                                                    SECH0003
    BEGIN REAL U;                                                              SECH0004
       U ← EXP (X);                                                            SECH0005
       SECH ← 2 / (U + 1 / U)                                                  SECH0006
    END SECH;                                                                  SECH0007

    COMMENT: READ INPUT - DATA CARDS, COMPUTE CORE AND CIRCUIT                 046
    PARAMETERS, AND PRINT HEADING.;                                            047
    STARTCORE : READ (CR, /, COREPARAMETERS) [QUIT];                           048
    STARTCKT : READ (CR, /, CIRCUITPARAMETERS) [QUITCKT];                      049
    V1 ← (PHIS - PHIR) / ((LO - LI) x HA);                                     050
    V2 ← (PHIS + PHIR) x HQ / ((LO - LI) x HN);                                051
    F12 ← HQ x LI;                                                             052
    F23 ← HQ x LO;                                                             053
    EPS ← V1 x LN (LO / LI);                                                   054
    NV ← 0.001 x EPS x SP;                                                     055
    ND ← 1;                                                                    056
    TR ← 2 x TM;                                                               057
    U ← 2 x SP / CAPID;                                                        058
    A1 ← (SP - CAPID / TM) / TM * 2;                                           059
    A2 ← (1.5 x CAPID / TM - SP) / TM;                                         060
    TS ← 900 x DELT;                                                           061
    TAUS ← PHIR / (0.3 x ROP x ND x CAPID);                                    062
    WRITE (F1 [PAGE]);                                                         063
    WRITE (F1, COREPARAMETERH, COREPARAMETERL);                                064
    WRITE (F1, CIRCUITPARAMETERH, CIRCUITPARAMETERL);                          065
    WRITE (F1, OUTPUTHEADING);                                                 066
    COMMENT: INITIALIZE VARIABLES AND PARAMETERS.;                             067
```

```
    T ← TIC ← 0;                                                              068
    ID ← F ← 0 ;                                                              069
    PHIC2 ← PHIC1 ← PHIC ← PHIDC ← -PHIR ;                                    070
    PHIDOTC1 ← PHIDOTC ← PHIDOTEC ←PHIDOTIC ← PHIDOTMAC ← 0 ;                 071
    LINES ←10;                                                                072
    K ←10;                                                                    073
    COUNT ← 0;                                                                074
    WRITE(F1,OUTPUTFORMAT,OUTPUTVARIABLES) ;                                  075
    COMMENT: COMPUTE VARIABLES DURING SWITCHING TIME.;                        076
    SWITCHING: T ← T + DELT;                                                  077
    COUNT ← COUNT + 1;                                                        078
    ID ← IF T < TM THEN T * 2 × (A1 × T + A2) ELSE                            079
         CAPID × (1 + TANH (U × (T - TM))) / 2;                               080
    F ← ND × ID;                                                              081
    FI ← FOI × TANH (F/FOI);                                                  082
    FDOTC ← IF T < TM THEN ND × T × (3 × A1 × T + 2 × A2) ELSE                083
       CAPID × ND × U × SECH (U × (T - TM)) * 2 / 2;                          084
    CTS ← 0;                                                                  085
    GUESS: PHIC ← PHIC2 + 2 × DELT × PHIDOTC1;                                086
    LOOP: CTS ← CTS + 1;                                                      087
    PHIDOTMAC ← PHIDOTMA (F, PHIC, PHIDC, PHIDOTMAPR);                        088
    PHIDOTEC ← PHIDOTE (FDOTC, DELT, NV, PHIDOTEPR);                          089
    PHIDOTIC ← PHIDOTI (F, T, TIC, PHIDOTIPR);                                090
    PHIDOTC ← PHIDOTMAC + PHIDOTEC + PHIDOTIC;                                091
    DELPHIC ← PHIC1 + DELT × (PHIDOTC + PHIDOTC1) / 2 - PHIC;                 092
    PHIC ← PHIC + DELPHIC;                                                    093
    IF CTS = 1 THEN C0 ← DELPHIC;                                             094
    IF CTS = 5 THEN                                                           095
    BEGIN THETA5 ← DELPHIC / C0;                                              096
       IF 0.9 < ABS (THETA5) THEN GO TO GUESS;                                097
       DELPHIC ← 0                                                            098
    END;                                                                      099
    IF ABS (DELPHIC) > 0.001× PHIR × DELT / TAUS AND CTS ≠ 6 THEN GO          100
    TO LOOP;                                                                  101
    PHIC2 ← PHIC1;                                                            102
    PHIC1 ← PHIC;                                                             103
    PHIDOTC1 ← PHIDOTC;                                                       104
    COMMENT: PRINT OUTPUT.;                                                   105
    IF P THEN K ← IF T < TR THEN 2 ELSE 10 ;                                  106
    IF COUNT MOD K  = 0 THEN                                                  107
    BEGIN                                                                     108
       IF LINES MOD 50 = 0 THEN                                               109
       BEGIN WRITE (F1 [PAGE]);                                               110
          WRITE (F1, OUTPUTHEADING)                                           111
       END;                                                                   112
       WRITE (F1, OUTPUTFORMAT, OUTPUTVARIABLES);                             113
       LINES ← LINES + 1                                                      114
    END;                                                                      115
    IF T ≤ TS THEN GO TO SWITCHING ;                                          116
    GO TO STARTCKT ;                                                          117
    QUITCKT : CLOSE (CR, SAVE);                                               118
    GO TO STARTCORE;                                                          119
    QUIT:                                                                     120
END.                                                                          121
```

A sample of input data:

```
  "E-6   ", 22.19Q-3,  23.54Q-3, 3.45Q-8, 3.726Q-8, 310.0, 35.0, 30.0,
0.069, 0.95, 1.30, 0.1132, 1.45, 3.12,0.01200,1.300,0.245Q-6,0.5500,
    0.60, 0.0592Q-6, 10.15Q6,0.4Q-9,          25Q6,  5,  400,  0,
    0.60, 0.0592Q-6, 10.15Q6,2Q-9,            5Q6,  5,  400, 1,
```

APPENDIX F

COMPUTER PROGRAM FOR LOADED CORE

Language: ALGOL 60

Program Description: Computes i_D, $\dot{\phi}$, ϕ, ϕ_d, F, $\dot{q}$, and V_d versus t for given core, circuit, and drive parameters.

Identifiers:

(1) *Analytical identifiers*

Identifier	Symbol	Identifier	Symbol
C	C	NU	ν
CAPID	I_D	PHIC	ϕ
DELQD	$\Delta\dot{q}$	PHIC1	ϕ_{n-1}
DELT	Δt	PHIC2	ϕ_{n-2}
DPHDTDFC	$\dot{\phi}'$	PHIDC	ϕ_d
ED	e_d	PHIDOTC	$\dot{\phi}$
EK	E_k	PHIDOTC1	$\dot{\phi}_{n-1}$
F	F	PHIR	ϕ_r
FB	F_B	PHIS	ϕ_s
FJ	$f_{n(j)}$	Q	q
FJPR	$f'_{n(j)}$	Q1	q_{n-1}
F0	F_0	QD	$\dot{q}$
F0PP	F''_0	QD1	$\dot{q}_{n-1}$
HA	H_a	QD2	$\dot{q}_{n-2}$
HN	H_n	QDD	$\ddot{q}$
HQ	H_q	QDD1	$\ddot{q}_{n-1}$
ID	i_D	R	R
I0	I_0	RD	R_d
L	L	RL	R_L
LAMBDA	λ	ROP	ρ_p
LI	l_i	T	t
LO	l_o	TAUS	τ_s
NC	N_c	TR	T_r
ND	N_D		

(2) *Auxiliary identifiers*

Identifier	Description
CIRCUITPARAMETERH	Format for the list CIRCUITPARAMETERL.
CIRCUITPARAMETERL	List of circuit parameters (N_c, R_L, L, C, R_d, I_0, E_k, I_D, N_D, and T_r).
CIRCUITPARAMETERS	List of circuit parameters (same as above) for input-data cards.
CORENAME	Core name, *e.g.* J-1.
COREPARAMETERH	Format for the list COREPARAMETERL.
COREPARAMETERL	List of core parameters (Core name, l_i, l_o, ϕ_r, ϕ_s, H_a, H_q, H_n, λ, F_0'', ν, ρ_p, F_0, and F_B).
COREPARAMETERS	List of core parameters (same as above) for input-data cards.
COUNT	Index number of nth Δt during switching.
CTS	Index number of jth iteration for each nth Δt.
GUESS	Label of location where initial approximation of ϕ is made for each nth Δt.
K	Index number of automatically plotted set of output.
LINES	Index number of printed line
LOOP	Label of location from where iterative computation is repeated for each nth Δt.
OUTPUTFORMAT	Format for the list OUTPUTVARIABLES.
OUTPUTHEADING	Format for output column heading.
OUTPUTVARIABLES	List of results (t, i_D, $\dot{\phi}$, ϕ, ϕ_d, F, $\dot{q}$, $e_d + \dot{q} R_d$, j_{max}).
P	$1/\ddot{q}_j$ if $\ddot{q}_j \neq 0$, zero otherwise.
P1	$1/(\dot{q}_{j-1} - \dot{q}_{n-1})$ if $\dot{q}_{j-1} \neq \dot{q}_{n-1}$, zero otherwise.
QUIT	Label of location where computation terminates
S	$1/C$ if C is finite, zero otherwise.
START	Label of location where computation starts for given core and circuit parameters.
SWITCHING	Label of location where computation starts for each nth Δt.
XSCALE	Time scale, used in automatic plotting of resulting waveforms.
YSCALE	$\dot{\phi}$ scale, used in automatic plotting of $\dot{\phi}(t)$.
ZSCALE	$\dot{q}$ scale, used in automatic plotting of $\dot{q}(t)$.

Program:

```
LOADED CORE                                                              001
BEGIN COMMENT: DECLARE CORE PARAMETERS, CIRCUIT PARAMETERS, VARIABLES    002
   , AND AUXILIARY IDENTIFIERS.;                                         003
   REAL LI, LO, PHIR, PHIS, HA, HQ, HN, LAMBDA, FOPP, NU, ROP, F0, FB    004
   ;                                                                     005
   REAL RL, L, C, RD, I0, TR, CAPID, EK, TAUS, R, S, P, P1;              006
   REAL T, DELT, PHIC, PHIC1, PHIC2, PHIDOTC, PHIDOTC1, ID, F, Q, Q1,    007
   QD, QD1, QD2, QDD, QDD1, DELQD, PHIDC, DPHDTDFC, DEPHIDOTC, DEQD,     008
   ED, FJ, FJPR, XSCALE, YSCALE, ZSCALE, G, G1;                          009
   INTEGER LINES, CTS, COUNT, NC, ND, K;                                 010
   ALPHA CORENAME;                                                       011
   LABEL START, SWITCHING, GUESS, LOOP, QUIT;                            012
   COMMENT: DECLARE INPUT / OUTPUT.;                                     013
   FILE IN CR 0(2, 10);                                                  014
   FILE OUT F1 4(2, 15);                                                 015
   LIST COREPARAMETERS(CORENAME, LI, LO, PHIR, PHIS, HA, HQ, HN,         016
   LAMBDA, FOPP, NU, ROP, F0, FB);                                       017
   LIST COREPARAMETERL(CORENAME, LI × 103, LO × 103, PHIR × 108, PHIS    018
   × 108, HA, HQ, HN, LAMBDA, FOPP, NU, ROP, F0, FB);                    019
   LIST CIRCUITPARAMETERS(NC, RL, L, C, RD, I0, EK, CAPID, ND, TR);      020
   LIST CIRCUITPARAMETERL(NC, RL, L × 106, C × 106, RD, I0 × 106, EK,    021
   CAPID, ND, TR × 106);                                                 022
   LIST OUTPUTVARIABLES(T × 106, ID, PHIDOTC, PHIC × 108, PHIDC × 108    023
   , F, QD, ED + QD × RD, CTS);                                          024
   FORMAT COREPARAMETERH("CORE ", A6, X4, "LI=", F8.3, X4, "LO=",        025
   F8.3, X4, "PHIR=", F8.3, X4, "PHIS=", F8.3, X4, "HA=", F8.3, X4,     026
   "HQ=", F8.3, X4, "HN=", F8.3 / X15, "LAMBDA=", F8.5, X4, "FOPP=",     027
   F8.3, X4, "NU=", F8.3, X4, "ROP=", F8.3, X4, "F0=", F8.3, X4,         028
   "FB=", F8.3 /);                                                       029
   FORMAT CIRCUITPARAMETERH("CIRCUIT", X8, "NC=", I2, X4, "RL=", F8.3    030
   , X4, "L=", F8.3, X4, "C=", F8.3, X4, "RD=", F8.3, X4, "I0=", F8.3    031
   , X4, "EK=", F8.5 / "PARAMETERS", X5, "CAPID=", F8.3, X4, "ND=",      032
   I2, X4, "TR=", F8.3 /);                                               033
   FORMAT OUTPUTHEADING(/ X6, "T", X11, "ID", X7, "PHIDOTC", X7,         034
   "PHIC", X7, "PHIDC", X9, "F", X11, "QD", X10, "VD", X9, "CTS" /);     035
   FORMAT OUTPUTFORMAT(F9.3, 2 F12.3, 2 F12.2, 3 F12.3, I10);            036
   COMMENT: PROCEDURES USED BY THIS PROGRAM ARE MIN, MAX, AND PHIDOT.    037
   ;                                                                     038
REAL PROCEDURE MIN(A,B);                                          MIN  01
   VALUE A,B; REAL A,B;                                           MIN  02
   BEGIN MIN ← IF A < B THEN A ELSE B END MIN ;                   MIN  03

REAL PROCEDURE MAX(A,B);                                          MAX  01
   VALUE A,B; REAL A,B;                                           MAX  02
   BEGIN  MAX ← IF A > B THEN A ELSE B END MAX ;                  MAX  03
```

PROCEDURE $\dot{\phi}(F, \phi, \phi_d, \dot{\phi}')$, APPENDIX D-1

```
   COMMENT: READ INPUT - DATA CARDS AND PRINT HEADINGS.;                 039
   READ(CR, /, COREPARAMETERS);                                          040
   START: READ(CR, /, CIRCUITPARAMETERS)[QUIT];                          041
   WRITE(F1[PAGE]);                                                      042
   WRITE(F1, COREPARAMETERH, COREPARAMETERL);                            043
   WRITE(F1, CIRCUITPARAMETERH, CIRCUITPARAMETERL);                      044
   WRITE(F1, OUTPUTHEADING);                                             045
```

```
LINES ← 6;                                                                    046
COUNT ← 0;                                                                    047
K ← 0;                                                                        048
COMMENT: INITIALIZE VARIABLES AND PARAMETERS.;                                049
PHIC2 ← PHIC1 ← - PHIR;                                                       050
QD2 ← QD1 ← QDD1 ← Q1 ← PHIDOTC1 ← 0;                                         051
S ← IF C < 1@10 THEN 1 / C ELSE 0;                                            052
R ← RL + RD;                                                                  053
TAUS ← IF TR ≤ 1@-7 THEN 2 × PHIR /(0.3 × ROP ×(ND × CAPID - F0PP)            054
) ELSE SQRT(PHIR × TR /(0.15 × ROP × ND × CAPID));                            055
DELT ← TAUS / 200;                                                            056
T ← TR × F0PP /(ND × CAPID);                                                  057
COMMENT: COMPUTE VARIABLES DURING SWITCHING TIME.;                            058
SWITCHING: T ← T + DELT;                                                      059
COUNT ← COUNT + 1;                                                            060
ID ← CAPID × MIN(T / TR, 1);                                                  061
CTS ← 0;                                                                      062
GUESS: PHIC ← PHIC2 + 2 × DELT × PHIDOTC1;                                    063
QD ← QD2 + 2 × DELT × QDD1;                                                   064
IF I0 < 1@10 THEN QD ← MAX(QD, 0);                                            065
Q ← Q1 + DELT ×(QD + QD1) / 2;                                                066
LOOP: G1 ← G;                                                                 067
CTS ← CTS + 1;                                                                068
F ← ND × ID - NC × QD;                                                        069
DEPHIDOTC ← PHIDOT(F, PHIC, PHIDC, DPHDTDFC) - PHIDOTC;                       070
PHIDOTC ← PHIDOTC + DEPHIDOTC;                                                071
PHIC ← PHIC1 + DELT ×(PHIDOTC + PHIDOTC1) / 2;                                072
ED ← EK × LN(QD / I0 + 1);                                                    073
IF L ≠ 0 THEN                                                                 074
BEGIN QDD ←(NC × PHIDOTC - Q / C - R × QD - ED) / L;                          075
   DEQD ← QD1 + DELT ×(QDD + QDD1) / 2 - QD;                                  076
END;                                                                          077
IF L = 0 THEN                                                                 078
BEGIN QDD ←(QD - QD1) / DELT;                                                 079
   FJ ← R × QD + ED + S × Q - NC × PHIDOTC;                                   080
   IF QDD ≠ 0 THEN P ← 1 / QDD ELSE P ← 0;                                    081
   FJPR ← R + EK /(QD + I0) + S × QD × DELT / 2 + DPHDTDFC × NC *             082
   2;                                                                         083
   DEQD ← - FJ / FJPR;                                                        084
END;                                                                          085
QD ← QD + DEQD;                                                               086
IF I0 < 1@10 THEN QD ← MAX(QD, 0);                                            087
G ← R × QD + ED + S × Q + L × QDD - NC × PHIDOTC;                             088
IF SIGN(G) ≠ SIGN(G1) THEN                                                    089
BEGIN QD ← QD - 0.5 × DEQD;                                                   090
   G ← G - 0.5 × DEQD × R;                                                    091
END;                                                                          092
Q ← Q1 + DELT ×(QD + QD1) / 2;                                                093
IF(ABS(DEQD) > 0.001 × QD AND CTS < 10) THEN GO TO LOOP;                      094
PHIC2 ← PHIC1;                                                                095
PHIC1 ← PHIC;                                                                 096
PHIDOTC1 ← PHIDOTC;                                                           097
QD2 ← QD1;                                                                    098
QD1 ← QD;                                                                     099
Q1 ← Q;                                                                       100
QDD1 ← QDD;                                                                   101
COMMENT  PRINT OUTPUT ;                                                       102
IF COUNT MOD 20 = 0 THEN                                                      103
BEGIN IF LINES MOD 50 = 0 THEN                                                104
   BEGIN WRITE(F1[PAGE]);                                                     105
      WRITE(F1, OUTPUTHEADING)                                                106
   END;                                                                       107
   WRITE(F1, OUTPUTFORMAT, OUTPUTVARIABLES);                                  108
   LINES ← LINES + 1                                                          109
```

```
END;
IF PHIDOTC ≠ 0 THEN GO TO SWITCHING;                                    110
GO TO START;                                                            111
QUIT: END.                                                              112
                                                                        113
```

A sample of input data:

```
  "J-1    ",7.18Ø-3,11.58Ø-3,31.0Ø-8,33.48Ø-8,250.0,26.0,22.50,1.64,
0.27,1.43,2.27,0.55,1.20,
2,1.58Ø,1.00Ø-6,1.000Ø20,0.00,1.0Ø20,0.0000,1.54,1,0.10Ø-6,          S--RL
2,0.131,0.38Ø-6,0.253Ø-6,0.00,1.0Ø20,0.0000,2.28,1,0.10Ø-6,          S--RLC
2,0.131,0.38Ø-6,0.253Ø-6,0.74,2.7Ø-6,0.0833,1.80,1,0.10Ø-6,          S--RLCD
```

APPENDIX G

COMPUTER PROGRAM FOR CORE-DIODE-TRANSISTOR BINARY COUNTER USING A SIMPLE METHOD OF SOLUTION

Language: ALGOL 60

Program Description: Computes i_L, i_s, F_1, ϕ_1, $\dot{\phi}_1$, F_2, ϕ_2, $\dot{\phi}_2$, i_d, V_d, $\dot{\phi}_{\epsilon 1}$, and $\dot{\phi}_{\epsilon 2}$ versus t.

Identifiers:

(1) *Analytical identifiers*

Identifier	Symbol	Identifier	Symbol
CAPIS	I_s	GJ	$g_{(j)}$
D	D	GJM1	$g_{(j-1)}$
DELID	δi_d	GPRID	$\partial g/\partial i_d$
DELIS	δi_s	GPRIS	$\partial g/\partial i_s$
DELT	Δt	HA	H_a
EK	E_k	HN	H_n
EPS	ϵ	HQ	H_q
FB	F_B	IC	I_c
FJ	$f_{(j)}$	ID	i_d
FJM1	$f_{(j-1)}$	IDM1	$i_{d(-1)}$
FPRID	$\partial f/\partial i_d$	IDM2	$i_{d(-2)}$
FPRIS	$\partial f/\partial i_s$	IL	i_L
F0	F_0	ILDOT	di_L/dt
F0PP	F_0''	ILDOTM1	$di_L/dt_{(-1)}$
F1	F_1	ILM1	$i_{L(-1)}$
F1DOT	$\dot{F}_1$	ILM2	$i_{L(-2)}$
F1M1	$F_{1(-1)}$	IS	i_s
F2	F_2	ISM1	$i_{s(-1)}$
F2DOT	$\dot{F}_2$	ISM2	$i_{s(-2)}$
F2M1	$F_{2(-1)}$	I0	I_0
F12	F_{12}	L	L
F23	F_{23}	LAMBDA	λ

Identifier	Symbol
LI	l_i
LO	l_o
NB1	N_{B1}
NB2	N_{B2}
NS1	N_{s1}
NS2	N_{s2}
NU	ν
PHIDOTE1	$\dot{\phi}_{\epsilon 1}$
PHIDOTE2	$\dot{\phi}_{\epsilon 2}$
PHIDOTEPR1	$\dot{\phi}'_{\epsilon 1}$
PHIDOTEPR2	$\dot{\phi}'_{\epsilon 2}$
PHIDOTMA1	$\dot{\phi}_{ma1}$
PHIDOTMA2	$\dot{\phi}_{ma2}$
PHIDOTMAPR1	$\dot{\phi}'_{ma1}$
PHIDOTMAPR2	$\dot{\phi}'_{ma2}$
PHIDOTPR1	$\dot{\phi}'_1$
PHIDOTPR2	$\dot{\phi}'_2$
PHIDOTSTR1	$\dot{\phi}^*_1$
PHIDOTSTR2	$\dot{\phi}^*_2$
PHIDOT1	$\dot{\phi}_1$
PHIDOT1M1	$\dot{\phi}_{1(-1)}$
PHIDOT2	$\dot{\phi}_2$
PHIDOT2M1	$\dot{\phi}_{2(-1)}$
PHIMA1	ϕ_{ma1}
PHIMA2	ϕ_{ma2}
PHIR	ϕ_r
PHIS	ϕ_s
PHI1	ϕ_1
PHI1M1	$\phi_{1(-1)}$
PHI1M2	$\phi_{1(-2)}$
PHI2	ϕ_2
PHI2M1	$\phi_{2(-1)}$
PHI2M2	$\phi_{2(-2)}$
RD	R_d
ROP	ρ_p
R1	R_1
R2	R_2
R3	R_3
R4	R_4
T	t
TAUS	τ_s
TIN	T_n
TI1	T_{i1}
TI2	T_{i2}
TR	T_r
V	V
VD	V_d
V1	V_1
V2	V_2

(2) *Auxiliary identifiers*

Identifier	Description
CF	Number of convergence failures.
CORE	Core name.
COUNT	Index number of Δt.
CTS	Index of jth iteration.
EXIT	Label of location where computation terminates.
FMT1	Format for the list LIST1.
FMT2	Format for the list LIST2.
FMT3	Format for the list LIST3.
FMT4	Format for output-column heading.

Identifier	Description
GUESS	Label location where predictions are made.
LINES	Index number of printed line.
LIST1	List of core parameters.
LIST2	List of circuit parameters.
LIST3	List of results (t, i_L, i_s, F_1, ϕ_1, $\dot{\phi}_1$, F_2, ϕ_2, $\dot{\phi}_2$, i_d, V_d, $\dot{\phi}_{\epsilon 1}$, $\dot{\phi}_{\epsilon 2}$, j_{max}, CF).
LOOP	Label of location where iterative computation begins.
NV	Negligible value of $\dot{\phi}_\epsilon$.
SWITCHING	Label of location where computation starts for each Δt.

Program:

```
BINARY COUNTER, USING SIMPLE INTEGRATION METHOD TO COMPUTE CURRENTS AND
VOLTAGES VS. TIME.
 BEGIN
 COMMENT: DECLARATION OF CORE PARAMETERS.;
 ALPHA CORE;
 REAL LI, LO, PHIR, PHIS, HA, HQ, HN, F0PP, FB, F0, ROP, NU, LAMBDA,
 F12, F23, V1, V2, EPS ;
 COMMENT: DECLARATION OF CIRCUIT PARAMETERS.;
 REAL R1, R2, R3, R4, L, NS1, NS2, NB1, NB2, EK, I0, RD, TR, IC, V,
 CAPIS, TAUS;
 COMMENT: DECLARATION OF VARIABLES.;
 REAL T, DELT, TI1, TI2, IL, ILM1, ILM2, IS, ISM1, ISM2, ID, IDM1,
 IDM2, VD, ILDOT, ILDOTM1, F1, F1M1, F2, F2M1, F1DOT, F2DOT, PHIDOT1,
 PHIDOTMA1, PHIDOT1M1, PHIDOT2, PHIDOTMA2,
 PHIDOTE1, PHIDOTEPR1, PHIDOTE2, PHIDOTEPR2,
 PHIDOT2M1, PHIDOTPR1, PHIDOTMAPR1, PHIDOTSTR1, PHIDOTPR2,
 PHIDOTMAPR2, PHIDOTSTR2, PHI1, PHIMA1,PHI1M1,
 PHI1M2, PHI2, PHIMA2, PHI2M1, PHI2M2, FJ, FJM1, FPRIS, FPRID,
 GJ, GJM1, GPRIS, GPRID, D, DELID, DELIS;
 COMMENT: MISCELLANEOUS DECLARATIONS.;
 REAL TIN, NV ;
 INTEGER LINES, COUNT, CTS, CF;
 LABEL SWITCHING, GUESS, LOOP, EXIT;
 COMMENT: DECLARATION OF FILE,OUTPUT LISTS AND FORMATS.;
 FILE OWT 4 (2, 15);
 LIST LIST1 (CORE, LI × 103, LO × 103, PHIR × 108, PHIS × 108, HA, HQ,
 HN, LAMBDA, F0PP, NU, ROP, F0, FB, EPS×109);
 LIST LIST2 (R1, R2, R3, R4, L × 103, NS1, NS2, NB1, NB2, EK, I0 × 106
 , RD, IC, V);
 LIST LIST3 (T × 106, IL, IS, F1, PHI1 × 108, PHIDOT1, F2, PHI2 × 108,
 PHIDOT2, ID, VD, PHIDOTE1, PHIDOTE2, CTS, CF);
 FORMAT FMT1 (X1, "CORE", A7, X4, "LI(MM)=", F8.3, X4, "LO(MM)=", F8.3
 , X4, "PHIR(MAXWELLS)=", F8.3, X4, "PHIS(MAXWELLS)=", F8.3 / X16, "HA(A
MP-TURNS/M)=", F8.3, X4, "HQ(AMP-TURNS/M)=", F8.3, X4, "HN(AMP-TURNS/M)=
", F8.3, X4, "LAMBDA=", F8.5 / X16, "F0PP(AMP-TURNS)=", F8.3, X4,
 "NU=", F8.3, X4, "RHOP(OHMS/TURN SQUARED)=", F8.3, X4,
 "F0(AMP-TURNS)=", F8.3 / X16, "FB(AMP-TURNS)=", F8.3, X4, "EPS(MILLIMIC
ROHENRYS/TURN SQUARED)=",F8.5/);
```

```
FORMAT FMT2 (X1, "CIRCUIT", X8, "R1(OHMS)=", F8.3, X4, "R2(OHMS)=",
F8.3, X4, "R3(OHMS)=", F8.3, X4, "R4(OHMS)=", F8.3 / X16, "L(MILLIHENRY
)=", F8.5, X4, "NS1=", I4, X4, "NS2=", I4, X4, "NB1=", I4, X4, "NB2=",
I4, X4, "EK(VOLT)=", F8.5 / X16, "IO(MICROAMP)=", F8.5, X4,
"RD(OHM)=", F8.3, X4, "IC(AMP)=", F8.3, X4, "V(VOLTS)=", F8.3 /);
FORMAT FMT3 (X1, F5.3, 3 F8.3, F9.3, 2 F8.3, F9.3, F8.3, F9.4, F9.3,
2 F9.4, 2 I5);
FORMAT FMT4 (X3, "T", X7, "IL", X6, "IS", X6, "F1", X5, "PHI1", X3, "PH
IDOT1", X4, "F2", X5, "PHI2", X3, "PHIDOT2", X4, "ID", X7, "VD", X4, "PH
IDTE1", X2, "PHIDTE2", X3, "CTS", X2, "CF" /);
COMMENT: DECLARATION OF PROCEDURES.;
```

PROCEDURE $\dot{\phi}_\epsilon(\dot{F}, \Delta t, NV, \dot{\phi}'_\epsilon)$, APPENDIX A
PROCEDURE $\dot{\phi}(F, \phi, \dot{\phi}', \dot{\phi}^*)$, APPENDIX D-2

```
COMMENT: INITIALIZE CORE PARAMETERS.;
CORE ← "100SC1";
HQ ← 42.7;
LAMBDA ← 0.640;
LI ← 5.590-3;
LO ← 7.980-3;
PHIR ← 6.250-8;
PHIS ← 7.000-8;
HA ← 290.0 ;
HN ← 38.0;
FOPP ← 0.35;
FB ← 3.00;
FO ← 0.805;
ROP ← 0.948;
NU ← 1.207;
F12 ← HQ × LI;
F23 ← HQ × LO;
V1 ← (PHIS - PHIR) / ((LO - LI) × HA);
V2 ← (PHIS + PHIR) × HQ / ((LO - LI) × HN);
EPS ← V1×LN(LO/LI) ;
COMMENT: INITIALIZE CIRCUIT PARAMETERS.;
TR ← 0.130Q-6;
R1 ← 107.36;
R2 ← 199.55;
R3 ← 0.340;
R4 ← 0.53;
L ← 0.2020-3;
NS1 ← 11;
NS2 ← 12;
NB1 ← 16;
NB2 ← 20.0;
EK ← 0.0578;
IO ← 0.0615Q-6;
RD ← 0.100;
NV ← 0.0001 ;
V ← 27.0;
COMMENT: ESTABLISH INITIAL CONDITIONS.;
T ← 0;
IL ← ISM2 ← ISM1 ← IS ← IDM2 ← IDM1 ← ID ← 0;
F1M1 ← F1 ← F2M1 ← F2 ← 0;
PHIDOT1 ← PHIDOTMA1 ← PHIDOTE1 ← PHIDOT2 ← PHIDOTMA2 ← PHIDOTE2 ← 0;
VD ← 0;
PHI1 ← PHI2 ← - PHIR;
```

```
CAPIS ← V / R1 × R2 / (R2 + 0.6 × ROP × (NS1 * 2 + NS2 * 2));                  093
TAUS ← 2.0 × PHIR × (NB1 * 2 + NB2 * 2) / (NB2 × 0.6 × ROP × (CAPIS ×          094
(NS1 × NB2 + NS2 × NB1) - FOPP × (NB1 + NB2)));                                095
IC ← V / R1;                                                                   096
CF ← 0;                                                                        097
WRITE (OWT [PAGE]);                                                            098
WRITE (OWT, FMT1, LIST1);                                                      099
WRITE (OWT, FMT2, LIST2);                                                      100
WRITE (OWT, FMT4);                                                             101
WRITE (OWT, FMT3, LIST3);                                                      102
LINES ← 12;                                                                    103
COUNT ← 0;                                                                     104
COMMENT: COMPUTE VARIABLES DURING SWITCHING TIME. ;                            105
SWITCHING:                                                                     106
BEGIN ILM2 ← ILM1;                                                             107
   ILM1 ← IL;                                                                  108
   PHI1M2 ← PHI1M1;                                                            109
   PHI1M1 ← PHI1;                                                              110
   PHI2M2 ← PHI2M1;                                                            111
   PHI2M1 ← PHI2;                                                              112
   PHIDOT1M1 ← PHIDOT1;                                                        113
   PHIDOT2M1 ← PHIDOT2;                                                        114
   ISM2 ← ISM1;                                                                115
   ISM1 ← IS;                                                                  116
   IDM2 ← IDM1;                                                                117
   IDM1 ← ID;                                                                  118
   ILDOTM1 ← ILDOT;                                                            119
   F1M1 ← F1;                                                                  120
   F2M1 ← F2;                                                                  121
   DELT ← IF T < TR THEN TAUS/1000 ELSE TAUS/500 ;                             122
   T ← T + DELT;                                                               123
   COUNT ← COUNT + 1;                                                          124
   CTS ← 0;                                                                    125
   GUESS: IF T ≤ TR THEN IL ← IC × (IF T < 0.0 THEN 0 ELSE IF T <              126
   0.020-6 THEN 1.02019 × T * 2.66667 ELSE IF T < 0.060-6 THEN 4.006           127
   × T - 0.05 ELSE IF T < 0.130-6 THEN 0.252 - 0.001 × (0.009470 + (           128
   TIN ← 13 - 108 × T) × (- 0.316910 + TIN × (1.729261 + TIN × (-              129
   0.575947 + TIN × 0.073769)))) ELSE 0.252) / 0.252 ELSE IL ← ILM2 +          130
  2.0 × DELT × ILDOTM1;                                                        131
  PHI1 ← PHI1M2 + 2.0 × DELT × PHIDOT1M1;                                      132
  PHI2 ← PHI2M2 + 2.0 × DELT × PHIDOT2M1;                                      133
  IS ← 2.0 × ISM1 - ISM2;                                                      134
  ID ← 2.0 × IDM1 - IDM2;                                                      135
  LOOP:                                                                        136
  FJM1 ← FJ;                                                                   137
  GJM1 ← GJ;                                                                   138
  CTS ← CTS + 1;                                                               139
  F1 ← NS1 × IS + NB1 × ID;                                                    140
  F2 ← NS2 × IS - NB2 × ID;                                                    141
  F1DOT ← (F1 - F1M1) / DELT;                                                  142
  F2DOT ← (F2 - F2M1) / DELT;                                                  143
  PHIDOTMA1 ← PHIDOT (F1, PHI1, PHIDOTMAPR1, PHIDOTSTR1);                      144
  PHIDOTE1 ← PHIDOTE (F1DOT,DELT,NV,PHIDOTEPR1) ;                              145
  PHIDOT1 ← PHIDOTMA1 + PHIDOTE1;                                              146
  PHIDOTPR1 ← PHIDOTMAPR1 + PHIDOTEPR1;                                        147
  PHIDOTMA2 ← PHIDOT (F2, PHI2, PHIDOTMAPR2, PHIDOTSTR2);                      148
  PHIDOTE2 ← PHIDOTE (F2DOT,DELT,NV,PHIDOTEPR2) ;                              149
  PHIDOT2 ← PHIDOTMA2 + PHIDOTE2;                                              150
  PHIDOTPR2 ← PHIDOTMAPR2 + PHIDOTEPR2;                                        151
  IF T > TR THEN                                                               152
  BEGIN ILDOT ← (- IL × R1 - NS1 × PHIDOT1 - NS2 × PHIDOT2 - IS × R3           153
     ) / L;                                                                    154
     IL ← ILM1 + 0.5 × DELT × (ILDOTM1 + ILDOT);                               155
  END;                                                                         156
```

```
   PHI1 ← PHI1M1 + 0.5 × DELT × (PHIDOT1M1 + PHIDOT1);                          157
   PHI2 ← PHI2M1 + 0.5 × DELT × (PHIDOT2M1 + PHIDOT2);                          158
   VD ← IF NB2 × PHIDOT2 > NB1 × PHIDOT1 THEN                                   159
   ID × RD + EK × LN (ID/I0 + 1.0 ) ELSE NB2 × PHIDOT2 - NB1 × PHIDOT1;         160
   FJ ← NB2 × PHIDOT2 - NB1 × PHIDOT1 - VD - ID × R4;                           161
   GJ ← NS2 × PHIDOT2 + NS1 × PHIDOT1 + IS × R3 - R2 × (IL - IS);               162
   FPRID ← - (NB2 * 2 × PHIDOTPR2 + NB1 * 2 × PHIDOTPR1 + RD + R4 +             163
                                                                                164
   EK / (ID + I0));                                                             165
   GPRIS ← NS2 * 2 × PHIDOTPR2 + NS1 * 2 × PHIDOTPR1 + R2 + R3;                 166
   FPRIS ← IF NB2 × PHIDOT2 > NB1 × PHIDOT1 THEN NS2 × NB2 × PHIDOTPR2          167
   - NS1 × NB1 × PHIDOTPR1 ELSE 0 ;                                             168
   GPRID ← - FPRIS;                                                             169
   D ← FPRID × GPRIS - FPRIS × GPRID;                                           170
   IF D ≠ 0 THEN                                                                171
   BEGIN DELID ← (- FJ × GPRIS + GJ × FPRIS) / D;                               172
      DELIS ← (FJ × GPRID - GJ × FPRID) / D;                                    173
   END ELSE DELID ← DELIS ← 0;                                                  174
   ID ← ID + DELID;                                                             175
   IS ← IS + DELIS;                                                             176
   IF SIGN (FJ) ≠ SIGN (FJM1) THEN ID ← ID - 0.5 × DELID;                       177
   IF ID ≤ 0 THEN ID ← 0 ;                                                      178
   IF SIGN (GJ) ≠ SIGN (GJM1) THEN IS ← IS - 0.5 × DELIS;                       179
   IF (ABS (DELID) > 0.0001 × ABS (ID) OR ABS (DELIS) > 0.0001 × ABS (          180
   IS)) AND CTS < 20 THEN GO TO LOOP;                                           181
   IF CTS = 20 THEN CF ← CF + 1;                                                182
   COMMENT: PRINT OUTPUT.;                                                      183
   IF COUNT MOD 20 = 0 THEN                                                     184
   BEGIN IF LINES MOD 50 = 0 THEN                                               185
      BEGIN WRITE (OWT [PAGE]);                                                 186
         WRITE (OWT, FMT4);                                                     187
      END;                                                                      188
      WRITE (OWT, FMT3, LIST3);                                                 189
      LINES ← LINES + 1;                                                        190
      CF ← 0;                                                                   191
   END;                                                                         192
   IF PHIDOTMA2 = 0 AND PHIDOTMA1 = 0 AND PHI1 > -0.9×PHIR THEN                 193
   GO TO EXIT ELSE GO TO SWITCHING;                                             194
END;                                                                            195
EXIT: END.
```

APPENDIX H

COMPUTER PROGRAM FOR CORE-DIODE-TRANSISTOR BINARY COUNTER USING THE RUNGE-KUTTA AND ADAMS METHODS OF SOLUTION

Language: ALGOL 60

Program Description: Computes i_L, i_s, F_1, ϕ_1, $\dot{\phi}_1$, F_2, ϕ_2, $\dot{\phi}_2$, i_d, V_d, $\dot{\phi}_{\epsilon 1}$, and $\dot{\phi}_{\epsilon 2}$ versus t.

Identifiers:

(1) *Analytical identifiers*

Identifier	Symbol	Identifier	Symbol
CAPIS	I_s	ID	i_d
DELT	Δt	IDM1	$i_{d(-1)}$
DELTMIN	Δt_{min}	IDM2	$i_{d(-2)}$
DELX	Δx	IL	i_L
DELXM1	$\Delta x_{(-1)}$	IS	i_s
EK	E_k	ISM1	$i_{s(-1)}$
EPS	ϵ	ISM2	$i_{s(-2)}$
FB	F_B	I0	I_0
FD1	$\dot{F}_1$	L	L
FD2	$\dot{F}_2$	LAMBDA	λ
F0	F_0	LI	l_i
F0PP	F_0''	LO	l_o
F1	F_1	NB1	N_{B1}
F1M1	$F_{1(-1)}$	NB2	N_{B2}
F2	F_2	NS1	N_{s1}
F2M1	$F_{2(-1)}$	NS2	N_{s2}
F12	F_{12}	NU	ν
F23	F_{23}	PHIDOTE1	$\dot{\phi}_{\epsilon 1}$
HA	H_a	PHIDOTE2	$\dot{\phi}_{\epsilon 2}$
HN	H_n	PHIDOTEPR1	$\dot{\phi}'_{\epsilon 1}$
HQ	H_q	PHIDOTEPR2	$\dot{\phi}'_{\epsilon 2}$
IC	I_c	PHIDOTMA1	$\dot{\phi}_{ma1}$

Identifier	Symbol	Identifier	Symbol
PHIDOTMA2	$\dot{\phi}_{ma2}$	ROP	ρ_p
PHIDOTMAPR1	$\dot{\phi}'_{maI}$	R1	R_1
PHIDOTMAPR2	$\dot{\phi}'_{ma2}$	R2	R_2
PHIDOTPR1	$\dot{\phi}'_1$	R3	R_3
PHIDOTPR2	$\dot{\phi}'_2$	R4	R_4
PHIDOTSTR1	$\dot{\phi}^*_1$	T	t
PHIDOTSTR2	$\dot{\phi}^*_2$	TAUS	τ_s
PHIDOT1	$\dot{\phi}_1$	TIN	T_n
PHIDOT2	$\dot{\phi}_2$	TR	T_r
PHIMA1	ϕ_{ma1}	TI1	T_{i1}
PHIMA2	ϕ_{ma2}	TI2	T_{i2}
PHIR	φ_r	V	V
PHIS	ϕ_s	VD	V_d
PHI1	ϕ_1	V1	V_1
PHI2	ϕ_2	V2	V_2
RD	R_d	XM1	$x_{(-1)}$

(2) *Auxiliary identifiers*

Identifier	Description
CORE	Core name.
COUNT	Index number of Δt.
EL	Lower limit of error in ADAMS PROCEDURE
EU	Upper limit of error in ADAMS PROCEDURE
EXIT	Label of location where computation terminates.
FCOUNT	Index number of jth iteration in *F* PROCEDURE.
FMT1	Format for the list LIST1.
FMT2	Format for the list LIST2.
FMT3	Format for the list LIST3.
FMT4	Format for output-column heading.
H	Array for $h(=\Delta x = \Delta t)$ to be used in *RK* and ADAMS PROCEDUREs.
I	Index for Δt to be used in *RK* and ADAMS PROCEDUREs.
LINES	Index number of printed line.
LIST1	List of core parameters.
LIST2	List of circuit parameters.

Identifier	Description
LIST3	List of results (t, i_L i_s, F_1, ϕ_1, $\dot{\phi}_1$, F_2, ϕ_2, $\dot{\phi}_2$, i_d, V_d, $\dot{\phi}_{\epsilon 1}$, $\dot{\phi}_{\epsilon 2}$, j_{max}, Δt).
NV	Negligible value of $\dot{\phi}_\epsilon$.
SWITCHING	Label of location where computation starts for each Δt.
X	Array for storing $x = t$.
Y	Array for storing a dependent variable.
YPR	Array for storing the time derivitive of a dependent variable.

Program:

```
BINARY COUNTER, USING RUNGE-KUTTA AND ADAMS METHODS TO COMPUTE CURRENTS
AND VOLTAGES VS. TIME.
 BEGIN
 COMMENT: DECLARATION OF CORE PARAMETERS.;
 ALPHA CORE;
 REAL LI, LO, PHIR, PHIS, HA, HQ, HN, F0PP, FB, F0, ROP, NU, LAMBDA,
 F12, F23, V1, V2, EPS;
 COMMENT: DECLARATION OF CIRCUIT PARAMETERS.;
 REAL R1, R2, R3, R4, L, NS1, NS2, NB1, NB2, EK, I0, RD, TR, IC, V,
 CAPIS, TAUS;
 COMMENT: DECLARATION OF VARIABLES.;
 REAL T, DELT, XM1, DELXM1, DELX, TI1, TI2, IL, ISM2, ISM1, IS, IDM2,
 IDM1, ID, VD, F1M1, F1, F2M1, F2, FD1, FD2, PHIDOT1, PHIDOTMA1,
 PHIDOTE1, PHIDOT2, PHIDOTMA2, PHIDOTE2, PHIDOTPR1, PHIDOTMAPR1,
 PHIDOTSTR1, PHIDOTEPR1, PHIDOTPR2, PHIDOTMAPR2, PHIDOTSTR2,
 PHIDOTEPR2, PHI1, PHIMA1, PHI2, PHIMA2;
 COMMENT: MISCELLANEOUS DECLARATIONS.;
 REAL EU, EL, DELTMIN, TIN, NV ;
 INTEGER I, LINES, COUNT, FCOUNT;
 REAL ARRAY H, X [0 : 5], Y, YPR [0 : 5, 0 : 3];
 LABEL SWITCHING, EXIT;
 COMMENT: DECLARATION OF FILE,OUTPUT LISTS AND FORMATS.;
 FILE OWT 4 (2, 15);
 LIST LIST1 (CORE, LI x 1Q3, LO x 1Q3, PHIR x 1Q8, PHIS x 1Q8, HA, HQ,
 HN, LAMBDA, F0PP, NU, ROP, F0, FB, EPSx1Q9);
 LIST LIST2 (R1, R2, R3, R4, L x 1Q3, NS1, NS2, NB1, NB2, EK, I0 x 1Q6
 , RD, IC, V);
 LIST LIST3 (T x 1Q6, IL, IS, F1, PHI1 x 1Q8, PHIDOT1, F2, PHI2 x 1Q8,
 PHIDOT2, ID, VD, PHIDOTE1, PHIDOTE2, FCOUNT,DELTx1Q9) ;
 FORMAT FMT1 (X1, "CORE", A7, X4, "LI(MM)=", F8.3, X4, "LO(MM)=", F8.3
 , X4, "PHIR(MAXWELLS)=", F8.3, X4, "PHIS(MAXWELLS)=", F8.3 / X16, "HA(A
MP-TURNS/M)=", F8.3, X4, "HQ(AMP-TURNS/M)=", F8.3, X4, "HN(AMP-TURNS/M)=
", F8.3, X4, "LAMBDA=", F8.5 / X16, "F0PP(AMP-TURNS)=", F8.3, X4,
 "NU=", F8.3, X4, "RHOP(OHMS/TURN SQUARED)=", F8.3, X4,
 "F0(AMP-TURNS)=", F8.3 / X16, "FB(AMP-TURNS)=", F8.3, X4, "EPS(MILLIMIC
ROHENRYS/TURN SQUARED)=",F8.5/);
 FORMAT FMT2 (X1, "CIRCUIT", X8, "R1(OHMS)=", F8.3, X4, "R2(OHMS)=",
 F8.3, X4, "R3(OHMS)=", F8.3, X4, "R4(OHMS)=", F8.3 / X16, "L(MILLIHENRY
)=", F8.5, X4, "NS1=", I4, X4, "NS2=", I4, X4, "NB1=", I4, X4, "NB2=",
 I4, X4, "EK(VOLT)=", F8.5 / X16, "I0(MICROAMP)=", F8.5, X4,
 "RD(OHM)=", F8.3, X4, "IC(AMP)=", F8.3, X4, "V(VOLTS)=", F8.3 /);
 FORMAT FMT3 (X1, F5.3, 3 F8.3, F9.3, 2 F8.3, F9.3, F8.3, F9.4, F9.3,
 2 F9.4, I3, F7.4) ;
```

```
FORMAT FMT4 (X3, "T", X7, "IL", X6, "IS", X6, "F1", X5, "PHI1", X3, "PH      044
IDOT1", X4, "F2", X5, "PHI2", X3, "PHIDOT2", X4, "ID", X7, "VD", X4, "PH      045
IDTE1", X2, "PHIDTE2"," CTS DELT"/);                                          046
COMMENT: DECLARATION OF PROCEDURES.;                                          047
```

PROCEDURE $\dot{\phi}_\epsilon(\dot{F}, \Delta t, NV, \dot{\phi}'_\epsilon)$, APPENDIX A
PROCEDURE $\dot{\phi}(F, \phi, \dot{\phi}', \dot{\phi}^*)$, APPENDIX D-2

```
                                                                              F     001
PROCEDURE F (X, Y, DX);                                                       F     002
VALUE X;                                                                      F     003
REAL X;                                                                       F     004
ARRAY Y [0], DX [0];                                                          F     005
COMMENT: THIS PROCEDURE USES THE FOLLOWING GLOBAL IDENTIFIERS: TR, R1         F     006
, R2, L, NS1, NS2, NB1, NB2, EK, I0, RD, IC, CAPIS, FCOUNT, T, XM1,           F     007
DELXM1, DELX, NV,  ISM2, ISM1, IS, IDM2, IDM1, ID, VD, F1M1, F1,              F     008
F2M1, F2, FD1, FD2, PHIDOT1, PHIDOTMA1, PHIDOTE1, PHIDOT2, PHIDOTMA2,         F     009
PHIDOTE2, PHIDOTPR1, PHIDOTMAPR1, PHIDOTSTR1, PHIDOTEPR1, PHIDOTPR2,          F     010
PHIDOTMAPR2, PHIDOTSTR2, PHIDOTEPR2;                                          F     011
BEGIN REAL U, V, UIS, UID, VIS, VID, NUMS, NUMD, DEEIS, DEEID, DENOM;         F     012
    LABEL ITERAT;                                                             F     013
    IF X ≠ XM1 THEN                                                           F     014
    BEGIN DELXM1 ← DELX;                                                      F     015
       DELX ← X - XM1;                                                        F     016
       IF DELXM1 = 0 THEN DELXM1 ← DELX;                                      F     017
       ISM2 ← ISM1;                                                           F     018
       ISM1 ← IS;                                                             F     019
       IDM2 ← IDM1;                                                           F     020
       IDM1 ← ID;                                                             F     021
       F1M1 ← F1;                                                             F     022
       F2M1 ← F2;                                                             F     023
       IS ← ISM1 + (ISM1 - ISM2) × DELX / DELXM1;                             F     024
       ID ← IDM1 + (IDM1 - IDM2) × DELX / DELXM1;                             F     025
    END;                                                                      F     026
    FCOUNT ← 0;                                                               F     027
    ITERAT : FCOUNT ← FCOUNT + 1;                                             F     028
    F1 ← NS1 × IS + NB1 × ID;                                                 F     029
    F2 ← NS2 × IS - NB2 × ID;                                                 F     030
    FD1 ← (F1 - F1M1) / DELX;                                                 F     031
    FD2 ← (F2 - F2M1) / DELX;                                                 F     032
    PHIDOTMA1 ← PHIDOT (F1, Y [1], PHIDOTMAPR1, PHIDOTSTR1);                  F     033
    PHIDOTE1 ← PHIDOTE (FD1, DELX, NV, PHIDOTEPR1) ;                          F     034
    DX [1] ← PHIDOT1 ← PHIDOTMA1 + PHIDOTE1;                                  F     035
    PHIDOTPR1 ← PHIDOTMAPR1 + PHIDOTEPR1;                                     F     036
    PHIDOTMA2 ← PHIDOT (F2, Y [2], PHIDOTMAPR2, PHIDOTSTR2);                  F     037
    PHIDOTE2 ← PHIDOTE (FD2, DELX, NV, PHIDOTEPR2) ;                          F     038
    DX [2] ← PHIDOT2 ← PHIDOTMA2 + PHIDOTE2;                                  F     039
    PHIDOTPR2 ← PHIDOTMAPR2 + PHIDOTEPR2;                                     F     040
    IF T ≤ TR THEN DX [3] ← IC × (IF T < 0.00 THEN 0 ELSE IF T <              F     041
    0.02@-6 THEN 2.72@19 × T * 1.66667 ELSE IF T < 0.06@-6 THEN 4.0@6         F     042
    ELSE 1.0@5 × (- 0.316910 + (TIN ← 13.0 - 1.0@8 × T) × (3.458522 +         F     043
    TIN × (- 1.727841 + TIN × 0.295076)))) / 0.252 ELSE DX [3] ← - (           F     044
    NS2 × PHIDOT2 + NS1 × PHIDOT1 + Y [3] × R1) / L;                          F     045
    VD ← IF NB2 × PHIDOT2 > NB1 × PHIDOT1 THEN                                F     046
    ID × RD + EK × LN (ID/I0 + 1.0 ) ELSE NB2 × PHIDOT2 - NB1 × PHIDOT1;      F     047
    U ← NS1 × PHIDOT1 + NS2 × PHIDOT2 + IS × R3 - (Y [3] - IS) × R2;          F     048
    V ← NB1 × PHIDOT1 - NB2 × PHIDOT2 + ID × R4 + VD;                         F     049
    UIS ← NS1 * 2 × PHIDOTPR1 + NS2 * 2 × PHIDOTPR2 + R2 + R3;                F     050
    VID ← NB1 * 2 × PHIDOTPR1 + NB2 * 2 × PHIDOTPR2 + RD + EK / (ID +         F     051
    I0) + R4;
```

```
    UID ← VIS ← IF NB2 × PHIDOT2 > NB1 × PHIDOT1 THEN                    F   052
    NS1 × NB1 × PHIDOTPR1 - NS2 × NB2 × PHIDOTPR2 ELSE 0 ;               F   053
    DENOM ← UIS × VID - UID × VIS;                                       F   054
    IF DENOM = 0 THEN DENOM ← 1@10;                                      F   055
    NUMS ← V × UID - U × VID;                                            F   056
    NUMD ← U × VIS - V × UIS;                                            F   057
    IF SIGN (NUMS / DENOM) ≠ SIGN (DEEIS) THEN DEEIS ← 0.5 × (NUMS /     F   058
    DENOM) ELSE DEEIS ← NUMS / DENOM;                                    F   059
    IF SIGN (NUMD / DENOM) ≠ SIGN (DEEID) THEN DEEID ← 0.5 × (NUMD /     F   060
    DENOM) ELSE DEEID ← NUMD / DENOM;                                    F   061
    IS ← IS + DEEIS;                                                     F   062
    ID ← ID + DEEID;                                                     F   063
    IF ID ≤ 0 THEN ID ← 0 ;                                              F   064
    IF FCOUNT < 10 AND (ABS (DEEIS) > 0.0001 × ABS (IS) OR ABS (DEEID)   F   065
    > 0.0001 × ABS (ID)) THEN GO TO ITERAT;                              F   066
    XM1 ← X;                                                             F   067
END OF F;                                                                F   068

PROCEDURE RKSTARTS (K, NF, X1, H, Y, YPR, F);                            RK  001
VALUE K, NF, H;                                                          RK  002
REAL X1, H;                                                              RK  003
INTEGER K, NF;                                                           RK  004
ARRAY Y, YPR [0, 0];                                                     RK  005
PROCEDURE F;                                                             RK  006
BEGIN INTEGER I, J;                                                      RK  007
   ARRAY DX, TEMPY, K1, K2, K3, K4, K5, K6 [0 : K];                      RK  008
   FOR I ← 0 STEP 1 UNTIL (NF - 1) DO                                    RK  009
   BEGIN                                                                 RK  010
      BEGIN REAL X;                                                      RK  011
         X ← H × I + X1;                                                 RK  012
         F (X, Y [I, *], DX);                                            RK  013
         FOR J ← 1 STEP 1 UNTIL K DO                                     RK  014
         BEGIN K1 [J] ← DX [J] × H;                                      RK  015
            TEMPY [J] ← K1 [J] / 3.0 + Y [I, J]                          RK  016
         END;                                                            RK  017
         F (H / 3.0 + X, TEMPY, DX);                                     RK  018
         FOR J ← 1 STEP 1 UNTIL K DO                                     RK  019
         BEGIN K2 [J] ← DX [J] × H;                                      RK  020
            TEMPY [J] ← (K2 [J] × 6.0 + K1 [J] × 4.0) / 25.0 + Y [I,     RK  021
            J]                                                           RK  022
         END;                                                            RK  023
         F ((H × 2.0) / 5.0 + X, TEMPY, DX);                             RK  024
         FOR J ← 1 STEP 1 UNTIL K DO                                     RK  025
         BEGIN K3 [J] ← DX [J] × H;                                      RK  026
            TEMPY [J] ← (K3 [J] × 15.0 - K2 [J] × 12.0 + K1 [J]) /       RK  027
            4.0 + Y [I, J]                                               RK  028
         END;                                                            RK  029
         F (H + X, TEMPY, DX);                                           RK  030
         FOR J ← 1 STEP 1 UNTIL K DO                                     RK  031
         BEGIN K4 [J] ← DX [J] × H;                                      RK  032
            TEMPY [J] ← (K4 [J] × 8.0 - K3 [J] × 50.0 + K2 [J] × 90.0    RK  033
            + K1 [J] × 6.0) / 81.0 + Y [I, J]                            RK  034
         END;                                                            RK  035
         F ((H × 2.0) / 3.0 + X, TEMPY, DX);                             RK  036
         FOR J ← 1 STEP 1 UNTIL K DO                                     RK  037
         BEGIN K5 [J] ← DX [J] × H;                                      RK  038
            TEMPY [J] ← (K4 [J] × 8.0 + K3 [J] × 10.0 + K2 [J] × 36.0    RK  039
            + K1 [J] × 6.0) / 75.0 + Y [I, J]                            RK  040
         END;                                                            RK  041
         F ((H × 4.0) / 5.0 + X, TEMPY, DX);                             RK  042
         FOR J ← 1 STEP 1 UNTIL K DO                                     RK  043
         BEGIN K6 [J] ← DX [J] × H;                                      RK  044
            Y [I + 1, J] ← (K1 [J] × 23.0 + K3 [J] × 125.0 - K5 [J] ×    RK  045
            81.0 + K6 [J] × 125.0) / 192.0 + Y [I, J]                    RK  046
```

```
                                                                        RK    047
      END;                                                              RK    048
    END XBLOCK;                                                         RK    049
    X [I + 1] ← X [I] + H;                                              RK    050
    F (X [I + 1], Y [I + 1, *], YPR [I + 1, *]);                        RK    051
    T ← X [I + 1];                                                      RK    052
    PHI1 ← Y [I + 1, 1];                                                RK    053
    PHI2 ← Y [I + 1, 2];                                                RK    054
    IL ← Y [I + 1, 3];                                                  RK    055
    PHIDOT1 ← YPR [I + 1, 1];                                           RK    056
    PHIDOT2 ← YPR [I + 1, 2];                                           RK    057
    F1 ← NS1 × IS + NB1 × ID;                                           RK    058
    F2 ← NS2 × IS - NB2 × ID;                                           RK    059
    VD ← ID × RD + EK × LN (ID / I0 + 1);                               RK    060
    WRITE (OWT, FMT3, LIST3);                                           RK    061
    LINES ← LINES + 1;                                                  RK    062
    COUNT ← COUNT + 1;                                                  RK    063
  END                                                                   RK    064
END PROCEDURE RKSTARTS;

PROCEDURE ADAMS (X, Y, YPRIME, N, EU, EL, EPS, H, HMIN);                ADAMS001
                                                                        ADAMS002
VALUE N;                                                                ADAMS003
ARRAY Y [0, 0], YPRIME [0, 0], X [0], H [0];                            ADAMS004
REAL EU, EL, EPS, HMIN;                                                 ADAMS005
INTEGER N;                                                              ADAMS006
BEGIN INTEGER I, J, Q;                                                  ADAMS007
  ALPHA B;                                                              ADAMS008
  REAL TEMP, KP, KC, KK, YC, E;                                         ADAMS009
  REAL ARRAY U [0 : 3], P [0 : 4], C [0 : 4], K [0 : 5], YP [0 : 20]    ADAMS010
  , FP [0 : 20];                                                        ADAMS011
  LABEL AWAY;                                                           ADAMS012
  FOR I ← 2 STEP 1 UNTIL 4 DO H [I] ← X [I] - X [I - 1];                ADAMS013
  H [5] ← H [4];                                                        ADAMS014
  IF B ≠ 0 THEN H [5] ← 2 × H [5];                                      ADAMS015
  B ← 0;                                                                ADAMS016
  FOR I ← 3 STEP 1 UNTIL 5 DO K [I] ← H [I - 1] / H [I];                ADAMS017
  AWAY: FOR I ← 1 STEP 1 UNTIL 3 DO                                     ADAMS018
  BEGIN U [I] ← 0;                                                      ADAMS019
    FOR J ← 5 - I STEP 1 UNTIL 4 DO U [I] ← K [J + 1] × (1 + U [I])     ADAMS020
  END;                                                                  ADAMS021
  P [1] ← - (3 + 4 × (U [1] + U [2]) + 6 × U [1] × U [2]) / (12 × U [   ADAMS022
  3] × (U [3] - U [1]) × (U [3] - U [2]));                              ADAMS023
  P [2] ← - (3 + 4 × (U [1] + U [3]) + 6 × U [1] × U [3]) / (12 × U [   ADAMS024
  2] × (U [2] - U [1]) × (U [2] - U [3]));                              ADAMS025
  P [3] ← - (3 + 4 × (U [2] + U [3]) + 6 × U [2] × U [3]) / (12 × U [   ADAMS026
  1] × (U [1] - U [2]) × (U [1] - U [3]));                              ADAMS027
  P [4] ← 1 - (P [1] + P [2] + P [3]);                                  ADAMS028
  C [1] ← (1 + 2 × U [1]) / (12 × U [2] × (U [2] + 1) × (U [2] - U [    ADAMS029
  1]));                                                                 ADAMS030
  C [2] ← (1 + 2 × U [2]) / (12 × U [1] × (U [1] + 1) × (U [1] - U [    ADAMS031
  2]));                                                                 ADAMS032
  C [4] ← (3 + 4 × (U [1] + U [2]) + 6 × U [1] × U [2]) / (12 × (U [    ADAMS033
  1] + 1) × (U [2] + 1));                                               ADAMS034
  C [3] ← 1 - (C [1] + C [2] + C [4]);                                  ADAMS035
  KP ← 0.2 - (P [3] × (U [1] * 4) + P [2] × (U [2] * 4) + P [1] × (U    ADAMS036
  [3] * 4));                                                            ADAMS037
  KC ← 0.2 - (C [4] + C [2] × U [1] * 4 + C [1] × (U [2] * 4));         ADAMS038
  KK ← KC / (KP - KC);                                                  ADAMS039
  FOR J ← 1 STEP 1 UNTIL N DO                                           ADAMS040
  BEGIN YP [J] ← Y [4, J] + H [5] × (P [4] × YPRIME [4, J] + P [3] ×    ADAMS041
    YPRIME [3, J] + P [2] × YPRIME [2, J] + P [1] × YPRIME [1, J]);     ADAMS042
    Y [5, J] ← YP [J]                                                   ADAMS043
  END;
```

```
    Q ← 0;                                                                    ADAMS044
    FOR J ← 1 STEP 1 UNTIL N DO                                               ADAMS045
    BEGIN I ← 5;                                                              ADAMS046
       X [5] ← X [4] + H [5];                                                 ADAMS047
       IF J = 1 THEN F (X [5], Y [5, *], FP);                                 ADAMS048
       YC ← Y [4, J] + H [5] × (C [4] × FP [J] + C [3] × YPRIME [4, J]        ADAMS049
       + C [2] × YPRIME [3, J] + C [1] × YPRIME [2, J]);                      ADAMS050
       YPRIME [5, J] ← FP [J];                                                ADAMS051
       E ← KK × (YC - YP [J]);                                                ADAMS052
       Y [5, J] ← YC + E;                                                     ADAMS053
       TEMP ← ABS (YC);                                                       ADAMS054
       IF TEMP < EPS THEN TEMP ← EPS;                                         ADAMS055
       IF ABS (E) > EU × TEMP AND H [5] ≥ HMIN × 2.0 THEN                     ADAMS056
       BEGIN H [5] ← 0.5 × H [5];                                             ADAMS057
          K [5] ← 2 × K [5];                                                  ADAMS058
          GO TO AWAY                                                          ADAMS059
       END;                                                                   ADAMS060
       TEMP ← ABS (YC);                                                       ADAMS061
       IF TEMP < EPS THEN TEMP ← EPS;                                         ADAMS062
       IF ABS (E) < EL × TEMP THEN Q ← Q + 1                                  ADAMS063
    END;                                                                      ADAMS064
    IF Q = N THEN B ← 1;                                                      ADAMS065
END ADAMS;                                                                    ADAMS066

STREAM PROCEDURE TRANSFER (N, A, B);                                          TRNSFR01
VALUE N;                                                                      TRNSFR02
BEGIN SI ← A;                                                                 TRNSFR03
   DI ← B;                                                                    TRNSFR04
   DS ← N WDS;                                                                TRNSFR05
END TRANSFER;                                                                 TRNSFR06

COMMENT: INITIALIZE CORE PARAMETERS.;                                           048
CORE ← "100SC1";                                                                049
HQ ← 42.7;                                                                      050
LAMBDA ← 0.640;                                                                 051
LI ← 5.590-3;                                                                   052
LO ← 7.980-3;                                                                   053
PHIR ← 6.250-8;                                                                 054
PHIS ← 7.000-8;                                                                 055
HA ← 290.0 ;                                                                    056
HN ← 38.0;                                                                      057
FOPP ← 0.35;                                                                    058
FB ← 3.00;                                                                      059
FO ← 0.805;                                                                     060
ROP ← 0.948;                                                                    061
NU ← 1.207;                                                                     062
F12 ← HQ × LI;                                                                  063
F23 ← HQ × LO;                                                                  064
V1 ← (PHIS - PHIR) / ((LO - LI) × HA);                                          065
V2 ← (PHIS + PHIR) × HQ / ((LO - LI) × HN);                                     066
EPS ← V1×LN(LO/LI) ;                                                            067
COMMENT: INITIALIZE CIRCUIT PARAMETERS.;                                        068
TR ← 0.1300-6;                                                                  069
R1 ← 107.36;                                                                    070
R2 ← 199.55;                                                                    071
R3 ← 0.340;                                                                     072
R4 ← 0.53;                                                                      073
L ← 0.2020-3;                                                                   074
NS1 ← 11;                                                                       075
NS2 ← 12;                                                                       076
NB1 ← 16;                                                                       077
NB2 ← 20.0;                                                                     078
EK ← 0.0578;                                                                    079
IO ← 0.06150-6;                                                                 080
```

```
RD ← 0.100;                                                                    081
V ← 27.0 ;                                                                     082
NV ← 0.0001 ;                                                                  083
COMMENT: ESTABLISH INITIAL CONDITIONS.;                                        084
T ← 0;                                                                         085
IL ← ISM2 ← ISM1 ← IS ← IDM2 ← IDM1 ← ID ← 0;                                  086
F1M1 ← F1 ← F2M1 ← F2 ← 0;                                                     087
PHIDOT1 ← PHIDOTMA1 ← PHIDOTE1 ← PHIDOT2 ← PHIDOTMA2 ← PHIDOTE2 ← 0;            088
VD ← 0;                                                                        089
PHI1 ← PHI2 ← - PHIR;                                                          090
CAPIS ← V / R1 × R2 / (R2 + 0.6 × ROP × (NS1 * 2 + NS2 * 2));                  091
TAUS ← 2.0 × PHIR × (NB1 * 2 + NB2 * 2) / (NB2 × 0.6 × ROP × (CAPIS ×          092
(NS1 × NB2 + NS2 × NB1) - FOPP × (NB1 + NB2)));                                093
IC ← V / R1;                                                                   094
WRITE (OWT [PAGE]);                                                            095
WRITE (OWT, FMT1, LIST1);                                                      096
WRITE (OWT, FMT2, LIST2);                                                      097
WRITE (OWT, FMT4);                                                             098
WRITE (OWT, FMT3, LIST3);                                                      099
LINES ← 12;                                                                    100
COUNT ← 0;                                                                     101
X [0] ← 0;                                                                     102
DELT ← TAUS / 500;                                                             103
DELTMIN ← DELT / 10;                                                           104
EU ← 0.001;                                                                    105
EL ← 0.0001;                                                                   106
Y [0, 1] ← Y [0, 2] ← - PHIR;                                                  107
Y [0, 3] ← 0;                                                                  108
XM1 ← - DELT;                                                                  109
RKSTARTS (3, 4, 0.0, DELT, Y, YPR, F);                                         110
SWITCHING: COUNT ← COUNT + 1;                                                  111
IF T ≤ TR THEN ADAMS (X, Y, YPR, 3, EU, EL, 0.1 × PHIR, H, DELTMIN)            112
ELSE ADAMS (X, Y, YPR, 3, 5.0 × EU, 10.0 × EL, 0.1 × PHIR, H, DELTMIN          113
);                                                                             114
DELT ← H [5];                                                                  115
T ← X [5];                                                                     116
PHI1 ← Y [5, 1];                                                               117
PHI2 ← Y [5, 2];                                                               118
IL ← Y [5, 3];                                                                 119
PHIDOT1 ← YPR [5, 1];                                                          120
PHIDOT2 ← YPR [5, 2];                                                          121
COMMENT: PRINT OUTPUT.;                                                        122
IF COUNT MOD 20 = 0 THEN                                                       123
BEGIN IF LINES MOD 50 = 0 THEN                                                 124
   BEGIN WRITE (OWT [PAGE]);                                                   125
      WRITE (OWT, FMT4);                                                       126
   END;                                                                        127
   WRITE (OWT, FMT3, LIST3);                                                   128
   LINES ← LINES + 1;                                                          129
END;                                                                           130
TRANSFER (4, X [2], X [1]);                                                    131
FOR I ← 2 STEP 1 UNTIL 5 DO                                                    132
BEGIN TRANSFER (3, Y [I, 1], Y [I - 1, 1]);                                    133
   TRANSFER (3, YPR [I, 1], YPR [I - 1, 1]);                                   134
END;                                                                           135
IF PHIDOTMA2 = 0 AND PHIDOTMA1 = 0 AND PHI1 > -0.9×PHIR THEN                   136
GO TO EXIT ELSE GO TO SWITCHING;                                               137
EXIT: END.                                                                     138
```

APPENDIX I

COMPUTER PROGRAM FOR FLUX DIVISION IN A LOADED SATURABLE CORE

Language: ALGOL 60

Program Description: Computes three types of output:

(1) $\dot{\phi}_m$, ϕ_m, F_m, $\dot{\phi}_3$, ϕ_3, F_3, $\dot{\phi}_4$, ϕ_4, F_4, and Ni versus t.

(2) D *vs.* NI and N_L^2/R_L for given l_4/l_3.

(3) D *vs.* l_4/l_3 and N_L^2/R_L for given NI.

Identifiers:

(1) *Analytical identifiers*

Identifier	Symbol	Identifier	Symbol
AM	A_m	FB4	F_{B4}
A3	A_3	FIM	F_{im}
A4	A_4	FI3	F_{i3}
BR	B_r	FI4	F_{i4}
BS	B_s	FJ	f
CIM	C_{im}	FJPR	f'
CI3	C_{i3}	FM	F_m
CI4	C_{i4}	F0IM	F_{0im}
D	D	F0I3	F_{0i3}
DF4	$\delta F_{4(j)}$	F0I4	F_{0i4}
DELF4	$F_4 - F_{4(-1)}$	F0M	F_{0m}
DELPHIM	$\phi_m - \phi_{m(-1)}$	F0 3	F_{03}
DELPHI3	$\phi_3 - \phi_{3(-1)}$	F0 4	F_{04}
DELPHI4	$\phi_4 - \phi_{4(-1)}$	F0PPM	F''_{0m}
DELT	Δt	F0PP 3	F''_{03}
DELTAPHI3	$\Delta\phi_3$	F0PP4	F''_{04}
DELTAPHI4	$\Delta\phi_4$	F12M	F_{12m}
FBM	F_{Bm}	F123	F_{123}
FB3	F_{B3}	F124	F_{124}

Identifier	Symbol
F23M	F_{23m}
F233	F_{233}
F234	F_{234}
F3	F_3
F4	F_4
F4M1	$F_{4(-1)}$
H	h
HA	H_a
HB	H_B
HN	H_n
HQ	H_q
H0	H_0
H0PP	H_0''
H0I	H_{0i}
KAPPA	κ
KAPPAI	κ_i
LAMBDAIM	λ_{im}
LAMBDAI3	λ_{i3}
LAMBDAI4	λ_{i4}
LAMBDAM	λ_m
LAMBDA3	λ_3
LAMBDA4	λ_4
LIM	l_{im}
LI3	l_{i3}
LI4	l_{i4}
LOM	l_{om}
LO3	l_{o3}
LO4	l_{o4}
LM	l_m
L3	l_3
L4	l_4
MI	M_i
NI	NI
NIV	Ni
NNR	N_L^2/R_L
NU	ν
NUI	ν_i

Identifier	Symbol
PHIDM	ϕ_{dm}
PHID3	ϕ_{d3}
PHID4	ϕ_{d4}
PHIDOTIPRM	$\dot{\phi}'_{im}$
PHIDOTIPR3	$\dot{\phi}'_{i3}$
PHIDOTIPR4	$\dot{\phi}'_{i4}$
PHIDOTM	$\dot{\phi}_m$
PHIDOTMAPRM	$\dot{\phi}'_{mam}$
PHIDOTMAPR3	$\dot{\phi}'_{ma3}$
PHIDOTMAPR4	$\dot{\phi}'_{ma4}$
PHIDOTMM1	$\dot{\phi}_{m(-1)}$
PHIDOT3M1	$\dot{\phi}_{3(-1)}$
PHIDOT4M1	$\dot{\phi}_{4(-1)}$
PHIDOTPRM	$\dot{\phi}'_m$
PHIDOTPR3	$\dot{\phi}'_3$
PHIDOTPR4	$\dot{\phi}'_4$
PHIDOT3	$\dot{\phi}_3$
PHIDOT4	$\dot{\phi}_4$
PHIM	ϕ_m
PHIMM1	$\phi_{m(-1)}$
PHIRM	ϕ_{rm}
PHIR3	ϕ_{r3}
PHIR4	ϕ_{r4}
PHISM	ϕ_{sm}
PHIS3	ϕ_{s3}
PHIS4	ϕ_{s4}
PHI3	ϕ_3
PHI3M1	$\phi_{3(-1)}$
PHI4	ϕ_4
PHI4M1	$\phi_{4(-1)}$
ROPM	ρ_{pm}
ROP3	ρ_{p3}
ROP4	ρ_{p4}
S	l_4/l_3
T	t
TAUS	τ_s
TIM	T_{im}

Identifier	Symbol	Identifier	Symbol
TR	T_r	V23	V_{23}
TI3	T_{i3}	V24	V_{24}
TI4	T_{i4}	ZETAP	ζ_p
V1M	V_{1m}	WM	w_m
V13	V_{13}	W3	w_3
V14	V_{14}	W4	w_4
V2M	V_{2m}		

(2) *Auxiliary identifiers*

Identifier	Description
ANS2	Array for storing D *vs.* NI for given N_L^2/R_L and l_4/l_3.
ANS3	Array for storing D *vs.* l_4/l_3 for given N_L^2/R_L and NI.
CF	Number of convergence failures for switching time.
CFS	Cumulative number of convergence failures.
COUNT	Index number of Δt.
CT	Index of jth iteration.
FMT1	Format for the list OUT1.
FMT2	Format for the list OUT2.
FMT3	Format for the list OUT3.
GUESS	Label for location where predictions are made.
I	Index number (general).
INI	Index for NI loop.
ILOAD	Index for N_L^2/R_L loop.
IS	Index for l_4/l_3 loop.
J	Index number (general).
LEGPARF	Format for the list LEGPARL
LEGPARL	List of leg parameters
LINES	Index number of printed line.
LOOP	Label of location where iterative computation begins.
MATPARF	Format for the list MATPARL.
MATPARL	List of material parameters.

Identifier	Description
OUT1	List of type-1 results.
SW	Number of output type (1, 2, or 3).
SWITCHING	Label of location where computation starts for each Δt.

(3) *DEFINE and PROCEDURE identifiers*

Identifier	Symbol	Identifier	Symbol
PHDOTI	$\dot{\phi}_i$	PHIDOTI4	$\dot{\phi}_{i4}$
PHDOTMA	$\dot{\phi}_{ma}$	PHIDOTMAM	$\dot{\phi}_{mam}$
PHIDOTIM	$\dot{\phi}_{im}$	PHIDOTMA3	$\dot{\phi}_{ma3}$
PHIDOTI3	$\dot{\phi}_{i3}$	PHIDOTMA4	$\dot{\phi}_{ma4}$

Program:

```
                                                                        001
FLUX DIVISION.                                                          002
   SW=1: FLUX SWITCHING IN EACH LEG VS. TIME FOR GIVEN LOAD, L4/L3,     003
         AND DRIVE;                                                     004
   SW=2: D VS. DRIVE AND LOAD FOR GIVEN L4/L3;                          005
   SW=3: D VS. L4/L3 AND LOAD FOR GIVEN DRIVE.                          006
 BEGIN COMMENT: DECLARATION OF MATERIAL PARAMETERS.;                    007
    REAL BR, BS, HA, HQ, HN, HOPP, NU, KAPPA, HO, HB, ZETAP, KAPPAI,    008
    NUI, MI, HOI;                                                       009
    COMMENT: DECLARATION OF LEG DIMENSIONS.;                            010
    REAL WM, AM, LIM, LOM, LM, W3, A3, LI3, LO3, L3, W4, A4, LI4, LO4,  011
    L4, H, S;                                                           012
    COMMENT: DECLARATION OF SWITCHING PARAMETERS.;                      013
    REAL PHIRM, PHISM, FOPPM, FBM, FOM, ROPM, LAMBDAM, F12M, F23M, V1M  014
    , V2M, LAMBDAIM, CIM, FOIM, FIM, PHIR3, PHIS3, FOPP3, FB3, FO3,     015
    ROP3, LAMBDA3, F123, F233, V13, V23, LAMBDAI3, CI3, FOI3, FI3,      016
    PHIR4, PHIS4, FOPP4, FB4, FO4, ROP4, LAMBDA4, F124, F234, V14, V24  017
    , LAMBDAI4, CI4, FOI4, FI4;                                         018
    COMMENT: DECLARATION OF CIRCUIT PARAMETERS.;                        019
    REAL NNR, TR, NI, NIV, TAUS;                                        020
    COMMENT: DECLARATION OF VARIABLES.;                                 021
    REAL T, DELT, FJ, FJPR, D, TIM, FM, PHIDOTM, PHIDOTMM1, PHIDOTPRM,  022
    PHIDOTMAPRM, PHIDOTIPRM, PHIM, PHIMM1, DELPHIM, PHIDM, TI3, F3,     023
    PHIDOT3, PHIDOT3M1, PHIDOTPR3, PHIDOTMAPR3, PHIDOTIPR3, PHI3,       024
    PHI3M1, DELPHI3, PHID3, DELTAPHI3, TI4, F4, PHIDOT4, PHIDOT4M1,     025
    PHIDOTPR4, PHIDOTMAPR4, PHIDOTIPR4, PHI4, PHI4M1, DELPHI4, PHID4,   026
    DELTAPHI4, F4M1, DELF4, DF4;                                        027
    COMMENT: DECLARATION OF MISCELLANEOUS.;                             028
    INTEGER CFS, CF, CT, COUNT, INI, ILOAD, IS, LINES, SW, I, J, K;     029
    LABEL SWITCHING, LOOP, GUESS;                                       030
    REAL ARRAY ANS2[0 : 20], ANS3[0 : 20, 0 : 20];                      031
    COMMENT: DECLARATION OF OUTPUT LISTS AND FORMATS;                   032
    FILE F1 4(2, 15);                                                   033
    LIST MATPARL(BR, BS, HA, HQ, HN, HOPP, NU, KAPPA, HO, HB, ZETAP,    034
    KAPPAI, NUI, MI x 1Q6, HOI);                                        035
    LIST LEGPARL(LIM x 1Q3, LOM x 1Q3, PHIRM x 1Q8, PHISM x 1Q8,        036
    LAMBDAM, FOPPM, ROPM, FOM, FBM, F12M, LAMBDAIM, CIM x 1Q6, FOIM,    037
    LI3 x 1Q3, LO3 x 1Q3, PHIR3 x 1Q8, PHIS3 x 1Q8, LAMBDA3, FOPP3,     038
    ROP3, FO3, FB3, F123, LAMBDAI3, CI3 x 1Q6, FOI3, LI4 x 1Q3, LO4 x
```

```
    103, PHIR4 x 108, PHIS4 x 108, LAMBDA4, F0PP4, ROP4, F04, FB4,
    F124, LAMBDAI4, CI4 x 106, F0I4);
    LIST OUT1(T x 106, PHIDOTM, PHIM x 108, FM, PHIDOT3, PHI3 x 108,
    F3, PHIDOT4, PHI4 x 108, F4, NIV, CT), OUT2(NNR, FOR J ← 1 STEP 1
    UNTIL 8 DO ANS2[J], CFS), OUT3(FOR I ← 1 STEP 1 UNTIL 6 DO FOR J ←
    0 STEP 1 UNTIL 9 DO ANS3[I, J]);
    FORMAT MATPARF(X40, "M A T E R I A L    P A R A M E T E R S" /
    "BR=", F6.3, X2, "BS=", F6.3, X2, "HA=", F5.1, X2, "HQ=", F4.1, X2
    , "HN=", F4.1, X2, "HOPP=", F4.1, X2, "NU=", F4.1, X2, "KAPPA=",
    F7.1, X2, "H0=", F4.1, X2, "HB=", F5.1, X2, "ZETAP=", F8.1 /
    "KAPPAI=", F6.1, X2, "NUI=", F4.2, X2, "MI=", F5.2, "Q-6", X2,
    "H0I=", F4.1 /);
    FORMAT LEGPARF(X40, "L E G    P A R A M E T E R S" / "LIM=", F6.3,
    X2, "LOM=", F6.3, X2, "PHIRM=", F6.3, X2, "PHISM=", F6.3, X2,
    "LAMBDAM=", F5.3, X2, "F0PPM=", F5.3, X2, "ROPM=", F5.3, X2,
    "F0M=", F5.3, X2, "FBM=", F5.3 / "F12M=", F5.3, X2, "LAMBDAIM=",
    F5.3, X2, "CIM=", F5.3, X2, "F0IM=", F5.3 / "LI3=", F6.3, X2,
    "L03=", F6.3, X2, "PHIR3=", F6.3, X2, "PHIS3=", F6.3, X2,
    "LAMBDA3=", F5.3, X2, "F0PP3=", F5.3, X2, "ROP3=", F5.3, X2,
    "F03=", F5.3, X2, "FB3=", F5.3 / "F123=", F5.3, X2, "LAMBDAI3=",
    F5.3, X2, "CI3=", F5.3, X2, "F0I3=", F5.3 / "LI4=", F6.3, X2,
    "L04=", F6.3, X2, "PHIR4=", F6.3, X2, "PHIS4=", F6.3, X2,
    "LAMBDA4=", F5.3, X2, "F0PP4=", F5.3, X2, "ROP4=", F5.3, X2,
    "F04=", F5.3, X2, "FB4=", F5.3 / "F124=", F5.3, X2, "LAMBDAI4=",
    F5.3, X2, "CI4=", F5.3, X2, "F0I4=", F5.3 /);
    FORMAT FMT1(11 F10.4, I6), FMT2(9 F11.4, I6), FMT3(9 F11.4, I6),
    TOP(X7, "T", X5, "PHIDOTM", X4, "PHIM", X8, "FM", X5, "PHIDOT3",
    X4, "PHI3", X8, "F3", X5, "PHIDOT4", X4, "PHI4", X8, "F4", X7,
    "NIV", X6 "CT"), HEAD1(X20,
    "FLUX SWITCHING IN EACH LEG FOR NL*2/RL =", F6.3, X5, "S =", F4.2,
    X5, "NI =", F4.2 / / / X7, "T", X5, "PHIDOTM", X4, "PHIM", X8,
    "FM", X5, "PHIDOT3", X4, "PHI3", X8, "F3", X5, "PHIDOT4", X4,
    "PHI4", X8, "F4", X7, "NIV", X6, "CT"), HEAD2(X35,
    "D VS. NI AND LOAD FOR S =", F4.2 / / / X4, "NL*2/RL", X40, "NI",
    X50, "CFS" / / X10, 8 F11.1), HEAD3(X35,
    "D VS. S AND LOAD FOR NI =", F4.0 / / / X4, "NL*2/RL", X40, "S",
    X50, "CFS" / / X10, 8 F11.1);
    COMMENT: DECLARATION OF DEFINITIONS AND PROCEDURES.;
```

```
DEFINE PHDOTI =
```

Lines PHDOTI07 through PHDOTI21
of
PROCEDURE $\dot{\phi}_i(F, t, T_i, \dot{\phi}'_i)$, APPENDIX B

```
  # ;
```

```
DEFINE PHDOTMA=
```

Lines PHDTMA06 through PHDTMA47
of
PROCEDURE $\dot{\phi}_{ma}(F, \phi, \phi_d, \dot{\phi}'_{ma})$, APPENDIX C

```
# ;

REAL PROCEDURE PHIDOTIM;                                                    PHDOTIM1
BEGIN DEFINE F = FM #, TI = TIM #, PHIDOTIPRIME = PHIDOTIPRM #,             PHDOTIM2
   PHIDOTI = PHIDOTIM #, LAMBDAI = LAMBDAIM #, CI = CIM #, FI =             PHDOTIM3
   FIM #;                                                                   PHDOTIM4
   PHDOTI                                                                   PHDOTIM5
END PHIDOTIM;                                                               PHDOTIM6

REAL PROCEDURE PHIDOTMAM;                                                   PHDTMAM1
BEGIN DEFINE F = FM #, PHI = PHIM #, PHID = PHIDM #, PHIDOTMAPRIME          PHDTMAM2
   = PHIDOTMAPRM #, PHIDOTMA = PHIDOTMAM #, LI = LIM #, LO = LOM #          PHDTMAM3
   , PHIR = PHIRM #, PHIS = PHISM #, LAMBDA = LAMBDAM #, FOPP =             PHDTMAM4
   FOPPM #, ROP = ROPM #, FO = FOM #, FB = FBM #, F12 = F12M #,             PHDTMAM5
   F23 = F23M #, V1 = V1M #, V2 = V2M #;                                    PHDTMAM6
   PHDOTMA                                                                  PHDTMAM7
END PHIDOTMAM;                                                              PHDTMAM8

REAL PROCEDURE PHIDOTI3;                                                    PHDOTI31
BEGIN DEFINE F = F3 #, TI = TI3 #, PHIDOTIPRIME = PHIDOTIPR3 #,             PHDOTI32
   PHIDOTI = PHIDOTI3 #, LAMBDAI = LAMBDAI3 #, CI = CI3 #, FI =             PHDOTI33
   FI3 #;                                                                   PHDOTI34
   PHDOTI                                                                   PHDOTI35
END PHIDOTI3;                                                               PHDOTI36

REAL PROCEDURE PHIDOTMA3;                                                   PHDTMA31
BEGIN DEFINE F = F3 #, PHI = PHI3 #, PHID = PHID3 #, PHIDOTMAPRIME          PHDTMA32
   = PHIDOTMAPR3 #, PHIDOTMA = PHIDOTMA3 #, LI = LI3 #, LO = LO3 #          PHDTMA33
   , PHIR = PHIR3 #, PHIS = PHIS3 #, LAMBDA = LAMBDA3 #, FOPP =             PHDTMA34
   FOPP3 #, ROP = ROP3 #, FO = FO3 #, FB = FB3 #, F12 = F123 #,             PHDTMA35
   F23 = F233 #, V1 = V13 #, V2 = V23 #;                                    PHDTMA36
   PHDOTMA                                                                  PHDTMA37
END PHIDOTMA3;                                                              PHDTMA38

REAL PROCEDURE PHIDOTI4;                                                    PHDOTI41
BEGIN DEFINE F = F4 #, TI = TI4 #, PHIDOTIPRIME = PHIDOTIPR4 #,             PHDOTI42
   PHIDOTI = PHIDOTI4 #, LAMBDAI = LAMBDAI4 #, CI = CI4 #, FI =             PHDOTI43
   FI4 #;                                                                   PHDOTI44
   PHDOTI                                                                   PHDOTI45
END PHIDOTI4;                                                               PHDOTI46

REAL PROCEDURE PHIDOTMA4;                                                   PHDTMA41
BEGIN DEFINE F = F4 #, PHI = PHI4 #, PHID = PHID4 #, PHIDOTMAPRIME          PHDTMA42
   = PHIDOTMAPR4 #, PHIDOTMA = PHIDOTMA4 #, LI = LI4 #, LO = LO4 #          PHDTMA43
   , PHIR = PHIR4 #, PHIS = PHIS4 #, LAMBDA = LAMBDA4 #, FOPP =             PHDTMA44
   FOPP4 #, ROP = ROP4 #, FO = FO4 #, FB = FB4 #, F12 = F124 #,             PHDTMA45
   F23 = F234 #, V1 = V14 #, V2 = V24 #;                                    PHDTMA46
   PHDOTMA                                                                  PHDTMA47
END PHIDOTMA4;                                                              PHDTMA48

REAL PROCEDURE TANH (X1);                                                   TANH0001
VALUE X1;                                                                   TANH0002
REAL X1;                                                                    TANH0003
BEGIN REAL Y;                                                               TANH0004
   Y ← EXP (X1 + X1);                                                       TANH0005
   TANH ← (Y - 1.0) / (Y + 1.0)                                             TANH0006
END TANH;                                                                   TANH0007

DEFINE MM = × 0.001 #;                                                          077
COMMENT: TYPE OF OUTPUT.;                                                       078
SW ← 2;                                                                         079
COMMENT: MATERIAL PARAMETERS.;                                                  080
BR ← 0.2300;                                                                    081
BS ← 1.08 × BR;                                                                 082
```

```
HA ← 310;                                                                    083
HQ ← 35.0;                                                                   084
HN ← 30.0;                                                                   085
HOPP ← 40.0;                                                                 086
NU ← 1.30;                                                                   087
KAPPA ← 3400;                                                                088
HO ← 61.0;                                                                   089
HB ←(HO × NU - HOPP) /(NU - 1);                                              090
ZETAP ← KAPPA × NU ×(HB - HOPP) *(NU - 1);                                   091
KAPPAI ← 592;                                                                092
NUI ← 1.3;                                                                   093
MI ← 10.7@-6;                                                                094
HOI ← 24.8;                                                                  095
COMMENT: CORE DIMENSIONS AND SWITCHING PARAMETERS.;                          096
H ← 1.31 MM;                                                                 097
WM ← 1.016 MM;                                                               098
AM ← H × WM;                                                                 099
LIM ← 14.363 MM;                                                             100
LOM ← 19.151 MM;                                                             101
LM ←(LIM + LOM) / 2;                                                         102
PHIRM ← AM × BR;                                                             103
PHISM ← AM × BS;                                                             104
FOPPM ← LM × HOPP;                                                           105
FBM ← LM × HB;                                                               106
FOM ← LM × HO;                                                               107
ROPM ← ZETAP × AM / LM;                                                      108
LAMBDAM ← KAPPA × AM / LM * NU;                                              109
F12M ← HQ × LIM;                                                             110
F23M ← HQ × LOM;                                                             111
V1M ←(PHISM - PHIRM) /(HA ×(LOM - LIM));                                     112
V2M ←(PHISM + PHIRM) × HQ /(HN ×(LOM - LIM));                                113
LAMBDAIM ← KAPPAI × AM / LM * NUI;                                           114
CIM ← MI × LM;                                                               115
FOIM ← HOI × LIM;                                                            116
W3 ← 0.508 MM;                                                               117
A3 ← H × W3;                                                                 118
LI3 ← 4.310 MM;                                                              119
LO3 ← 5.108 MM;                                                              120
L3 ←(LI3 + LO3) / 2;                                                         121
PHIR3 ← A3 × BR;                                                             122
PHIS3 ← A3 × BS;                                                             123
FOPP3 ← L3 × HOPP;                                                           124
FB3 ← L3 × HB;                                                               125
FO3 ← L3 × HO;                                                               126
ROP3 ← ZETAP × A3 / L3;                                                      127
LAMBDA3 ← KAPPA × A3 / L3 * NU;                                              128
F123 ← HQ × LI3;                                                             129
F233 ← HQ × LO3;                                                             130
V13 ←(PHIS3 - PHIR3) /(HA ×(LO3 - LI3));                                     131
V23 ←(PHIS3 + PHIR3) × HQ /(HN ×(LO3 - LI3));                                132
LAMBDAI3 ← KAPPAI × A3 / L3 * NUI;                                           133
CI3 ← MI × L3;                                                               134
FOI3 ← HOI × LI3;                                                            135
W4 ← 0.508 MM;                                                               136
A4 ← H × W4;                                                                 137
PHIR4 ← A4 × BR;                                                             138
PHIS4 ← A4 × BS;                                                             139
IF SW = 3 THEN                                                               140
BEGIN WRITE(F1, MATPARF, MATPARL);                                           141
   WRITE(F1, HEAD3, 100.0, FOR S ← 1 STEP 2 UNTIL 9 DO S);                   142
   LINES ← 7;                                                                143
END;                                                                         144
IS ← 0;                                                                      145
FOR S ← 1.83 DO                                                              146
```

```
BEGIN IS ← IS + 1;                                                    147
   LI4 ← S × LI3;                                                     148
   LO4 ← S × LO3;                                                     149
   L4 ←(LI4 + LO4) / 2;                                               150
   FOPP4 ← L4 × HOPP;                                                 151
   FB4 ← L4 × HB;                                                     152
   FO4 ← L4 × HO;                                                     153
   ROP4 ← ZETAP × A4 / L4;                                            154
   LAMBDA4 ← KAPPA × A4 / L4 * NU;                                    155
   F124 ← HQ × LI4;                                                   156
   F234 ← HQ × LO4;                                                   157
   V14 ←(PHIS4 - PHIR4) /(HA ×(LO4 - LI4));                           158
   V24 ←(PHIS4 + PHIR4) × HQ /(HN ×(LO4 - LI4));                      159
   LAMBDAI4 ← KAPPAI × A4 / L4 * NUI;                                 160
   CI4 ← MI × L4;                                                     161
   FOI4 ← HOI × LI4;                                                  162
   IF SW = 2 THEN                                                     163
   BEGIN WRITE(F1[PAGE]);                                             164
      WRITE(F1, MATPARF, MATPARL);                                    165
      WRITE(F1, LEGPARF, LEGPARL);                                    166
      WRITE(F1, HEAD2, S, FOR NI ← 1.1, 1.3, 1.5, 1.7, 2.0, 3.0,      167
      4.0, 5.0 DO NI);                                                168
      LINES ← 16;                                                     169
   END;                                                               170
   COMMENT: CIRCUIT PARAMETERS.;                                      171
   TR ← 0.08@-6;                                                      172
   ILOAD ← 0;                                                         173
   FOR NNR ← 0, 1.0, 3.962, 9.524 DO                                  174
   BEGIN ILOAD ← ILOAD + 1;                                           175
      CFS ← 0;                                                        176
      INI ← 0;                                                        177
      FOR NI ← 1.1, 1.3, 1.5, 1.7, 2.0, 3.0, 4.0, 5.0 DO              178
      BEGIN INI ← INI + 1;                                            179
         IF SW = 1 THEN                                               180
         BEGIN WRITE(F1[PAGE]);                                       181
            WRITE(F1, MATPARF, MATPARL);                              182
            WRITE(F1, LEGPARF, LEGPARL);                              183
            WRITE(F1, HEAD1, NNR, S, NI);                             184
            LINES ← 16;                                               185
         END;                                                         186
         COMMENT: INITIAL CONDITIONS.;                                187
         T ← 0;                                                       188
         PHIM ← - PHIRM;                                              189
         PHI3 ← - PHIR3;                                              190
         PHI4 ← - PHIR4;                                              191
         FM ← F3 ← F4 ← 0;                                            192
         CF ← COUNT ← 0;                                              193
         PHIDOTM ← PHIDOT3 ← PHIDOT4 ← 0;                             194
         COMMENT: COMPUTATION OF SWITCHING.;                          195
         TAUS ← 1.65 × PHIRM × ((1 / LAMBDAM *(1 / NU) + 1 / LAMBDA3  196
         *(1 / NU)) /(NI - FOPPM - FOPP3)) * NU;                      197
         DELT ← TAUS / 200;                                           198
         SWITCHING: CT ← 0;                                           199
         T ← T + DELT;                                                200
         NIV ← NI ×(IF T < TR THEN(1 - COS(3.141592654 × T / TR))     201
         / 2 ELSE 1);                                                 202
         PHIMM1 ← PHIM;                                               203
         PHIDOTMM1 ← PHIDOTM;                                         204
         PHI3M1 ← PHI3;                                               205
         PHIDOT3M1 ← PHIDOT3;                                         206
         PHI4M1 ← PHI4;                                               207
         PHIDOT4M1 ← PHIDOT4;                                         208
         F4M1 ← F4;                                                   209
         GUESS: PHIM ← PHIMM1 + DELPHIM;                              210
```

```
            PHI3 ← PHI3M1 + DELPHI3;                                          211
            PHI4 ← PHI4M1 + DELPHI4;                                          212
            F4 ← F4M1 + DELF4;                                                213
            LOOP: CT ← CT + 1;                                                214
            F4 ← F4 + DF4;                                                    215
            FI4 ← FOI4 × TANH(F4 / FOI4);                                     216
            PHIDOT4 ← PHIDOTMA4 + PHIDOTI4;                                   217
            DELPHI4 ← DELT ×(PHIDOT4 + PHIDOT4M1) / 2;                        218
            PHI4 ← PHI4M1 + DELPHI4;                                          219
            F3 ← F4 + NNR × PHIDOT4;                                          220
            FI3 ← FOI3 × TANH(F3 / FOI3);                                     221
            PHIDOT3 ← PHIDOTMA3 + PHIDOTI3;                                   222
            DELPHI3 ← DELT ×(PHIDOT3 + PHIDOT3M1) / 2;                        223
            PHI3 ← PHI3M1 + DELPHI3;                                          224
            FM ← NIV - F3;                                                    225
            FIM ← FOIM × TANH(FM / FOIM);                                     226
            PHIDOTM ← PHIDOTMAM + PHIDOTIM;                                   227
            DELPHIM ← DELT ×(PHIDOTM + PHIDOTMM1) / 2;                        228
            PHIM ← PHIMM1 + DELPHIM;                                          229
            DELF4 ← F4 - F4M1;                                                230
            FJ ← PHIDOT4 + PHIDOT3 - PHIDOTM;                                 231
            PHIDOTPRM ← PHIDOTIPRM + PHIDOTMAPRM;                             232
            PHIDOTPR3 ← PHIDOTIPR3 + PHIDOTMAPR3;                             233
            PHIDOTPR4 ← PHIDOTIPR4 + PHIDOTMAPR4;                             234
            FJPR ←(PHIDOTPRM + PHIDOTPR3) ×(1 + NNR × PHIDOTPR4) +            235
            PHIDOTPR4;                                                        236
            IF FJPR = 0 THEN FJPR ← 1@-20;                                    237
            DF4 ← IF SIGN(DF4) = SIGN(- FJ / FJPR) THEN - FJ / FJPR           238
            ELSE - 0.5 × FJ / FJPR;                                           239
            IF ABS(FJ) > 0.0001 × PHIRM / TAUS AND CT < 20 THEN GO TO         240
            LOOP;                                                             241
            IF CT = 20 THEN CF ← CF + 1;                                      242
            IF COUNT MOD 5 = 0 AND SW = 1 THEN                                243
            BEGIN WRITE(F1, FMT1, OUT1);                                      244
               LINES ← LINES + 1;                                             245
               IF LINES MOD 50 = 0 THEN                                       246
               BEGIN WRITE(F1[PAGE]);                                         247
                  WRITE(F1, TOP);                                             248
               END;                                                           249
            END;                                                              250
            COUNT ← COUNT + 1;                                                251
            IF PHIM < 0 AND PHIDOTM > 0.0001 OR PHIM < - 0.99 × PHIRM          252
            THEN GO TO SWITCHING;                                             253
            DELTAPHI3 ← PHI3 + PHIR3;                                         254
            DELTAPHI4 ← PHI4 + PHIR4;                                         255
            IF DELTAPHI4 = 0 THEN D ← 1@20 ELSE D ← DELTAPHI3 /               256
            DELTAPHI4;                                                        257
            CFS ← CFS + CF;                                                   258
            IF SW = 2 THEN ANS2[INI] ← D;                                     259
            IF SW = 3 THEN ANS3[ILOAD, IS] ← D;                               260
         END NILOOP;                                                          261
         IF SW = 2 THEN WRITE(F1, FMT2, OUT2);                                262
         IF SW = 3 THEN                                                       263
         BEGIN ANS3[ILOAD, 0] ← NNR;                                          264
            ANS3[ILOAD, 9] ← ANS3[ILOAD, 9] + CFS                             265
         END;                                                                 266
      END LOADLOOP;                                                           267
   END SLOOP;                                                                 268
   IF SW = 3 THEN WRITE(F1, FMT3, OUT3);                                      269
   END.                                                                       270
```

REFERENCES

1. D. Nitzan, "Flux Switching in Multipath Cores," Report 1, for Jet Propulsion Laboratory, Contract 950095 under NASw-6, SRI Project 3696, Stanford Research Institute, Menlo Park, California (November 1961).

2. D. Nitzan and V. W. Hesterman, "Flux Switching in Multipath Cores," Report 2, for Jet Propulsion Laboratory, Contract 950095 under NASw-6, SRI Project 3696, Stanford Research Institute, Menlo Park, California (November 1962).

3. D. Nitzan and V. W. Hesterman, "Flux Switching in Multipath Cores," Report 3, for Jet Propulsion Laboratory, Contract 950095 under NASw-6, SRI Project 3696, Stanford Research Institute, Menlo Park, California (June 1964).

4. N. Cushman and D. Park, "Reversal of a Loaded Ferromagnetic Core," *IRE Trans. on Component Parts*, vol. CP-7, pp. 117-124 (December 1960).

5. R. W. McKay, "The Reversible Component of Magnetization," *J. Appl. Phys.*, Supplement to vol. 30.4, pp. 56S-57S (April 1959).

6. R. L. Conger and F. C. Essig, "Magnetization Reversal in Thin Films at Low Fields," *J. Appl. Phys.*, vol. 28, pp. 855-858 (August 1957).

7. W. E. Milne, *Numerical Solution of Differential Equations* (John Wiley & Sons, Inc., New York, 1950).

8. P. Henrici, *Elements of Numerical Analysis* (John Wiley & Sons, Inc., New York, 1963), Chapter 4.

9. J. B. Scarborough, *Numerical Mathematical Analysis* (The Johns Hopkins Press, 1950), Chapter 9.

10. A. F. Pixley and Anne G. Macek, "A B5000 ALGOL-60 Runge-Kutta-Nystrøm Starting Procedure for Use with Milne Type Predictor-Connectors in Solving Systems of Ordinary Differential Equations," Burroughs Technical Bulletin MRS-113 (August 1, 1963).

11. PROCEDURE ADAMS (X, Y, YPRIME, N, EU, EL, EPS, H, HMIN, DERIV) in the computer-program library at Stanford Research Institute, Menlo Park, California.

12. D. Nitzan, "Flux Division in a Saturable Multipath Core," *IEEE Trans. on Electronic Computers*, vol. EC-13, pp. 272-277 (June 1964).

13. "Burroughs B5500 Information Processing System Extended ALGOL Reference Manual," Burroughs Corporation, Detroit, Michigan (1964).

14. N. Menyuk and J. B. Goodenough, "Magnetic Materials for Digital-Computer Components. I. A Theory of Flux Reversal in Polycrystalline Ferromagnetics," *J. Appl. Phys.*, vol. 26, pp. 8-18 (January 1955).

INDEX